

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-100861-8785

**INTELIGENTNÉ METÓDY DIAGNOSTIKY A RIADENIA
MECHATRONICKÝCH SYSTÉMOV**

DIZERTAČNÁ PRÁCA

2020

Ing. Zuzana Képešiová

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-100861-8785

**INTELIGENTNÉ METÓDY DIAGNOSTIKY A RIADENIA
MECHATRONICKÝCH SYSTÉMOV**

DIZERTAČNÁ PRÁCA

Študijný program:

Mechatronické systémy

Názov študijného odboru:

Kybernetika

Školiace pracovisko:

Ústav automobilovej mechatroniky

Vedúci záverečnej práce:

prof. Ing. Danica Rosinová, PhD.

Konzultant:

prof. Ing. Štefan Kozák, PhD.

2020

Ing. Zuzana Képešiová

SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program:	Mechatronické systémy
Názov študijného odboru:	Kybernetika
Dizertačná práca:	Inteligentné metódy diagnostiky a riadenia mechatronických systémov
Autor:	Ing. Zuzana Képešiová
Vedúci záverečnej práce:	prof. Ing. Danica Rosinová, PhD.
Konzultant:	prof. Ing. Štefan Kozák, PhD.
Miesto a rok predloženia práce:	Bratislava 2020

Mechatronické systémy a zariadenia sú základnou zložkou priemyslu modernej doby, najnovšieho konceptu - Priemysel 4.0 (Industry 4.0). Priemysel 4.0. spája v sebe multidisciplinárne využitie širokého spektra vedných a aplikačných oblastí tak, aby bolo možné integrovať najnovšie poznatky z metód a algoritmov automatického riadenia, elektroniky, informačno-komunikačných technológií a umelej inteligencie do nových výrobkov, technológií a priemyselných procesov. Výsledkom a hlavným prínosom dizertačnej práce je autorkou navrhnutý, spracovaný a implementovaný moderný prístup k diagnostike a riadeniu mechatronických zariadení v sieti IIoT pre splnenie potrieb trhu po inovatívnych riešeniacach spĺňajúcich koncept priemyselnej revolúcie Priemysel 4.0. Informácie o stave a prevádzke zariadenia sú automaticky posielané do výpočtovej jednotky umiestnenej v cloude. V cloude sú tieto dátá automaticky spracované umelou inteligenciou a je vyhodnotený stav sledovaného systému na báze určenia prípadnej chyby.

Kľúčové slová: mechatronické systémy, Priemysel 4.0, umelá inteligencia, diagnostické postupy

ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION
TECHNOLOGY

Study Programme:	Mechatronic Systems
Study Branch:	Cybernetics
Dissertation Thesis:	Intelligent methods of diagnostics and control of mechatronic systems
Author:	Ing. Zuzana Képešiová
Supervisor:	prof. Ing. Danica Rosinová, PhD.
Consultant:	prof. Ing. Štefan Kozák, PhD.
Place and year of submission:	Bratislava 2020

Mechatronic systems and devices are a basic component of the modern industry, the latest concept - Industry 4.0. Industry 4.0. combines the multidisciplinary use of a wide range of scientific and application areas so that it is possible to integrate the latest knowledge from methods and algorithms of automatic control, electronics, information and communication technologies and artificial intelligence into new products, technologies and industrial processes. The result of the thesis is the author's design, development and implementation of a modern approach to diagnostics and control of mechatronic devices in the IIoT network to meet the needs of the market by providing innovative solutions fulfilling the concept of the latest industrial revolution Industry 4.0. Information about the device status and functioning is automatically sent to the computing unit located in the cloud. In the cloud, this data is automatically processed by artificial intelligence and the device's health is evaluated, a fault is detected and classified.

Key words: mechatronic systems, Industry 4.0, artificial intelligence, diagnostic practices

Vyhľásenie autora

Podpísaná Ing. Zuzana Képešiová čestne vyhlasujem, že som dizertačnú prácu Inteligentné metódy diagnostiky a riadenia mechatronických systémov vypracovala na základe poznatkov získaných počas štúdia a informácií z dostupnej literatúry uvedenej v práci.

Uvedenú prácu som vypracovala pod vedením prof. Ing. Danice Rosinovej, PhD., odborným konzultantom bol prof. Ing. Štefan Kozák, PhD.

V Bratislave dňa 31.8.2020

.....
podpis autora

Pod'akovanie

Touto cestou by som rada pod'akovala pánovi prof. Ing. Štefanovi Kozákovi, PhD., ktorý sa venoval potrebám svojich študentov a nasmeroval ich na správnu cestu, tak ako aj mňa pri výbere vhodnej témy a uvedenia do problematiky. Rada by som pod'akovala aj vedúcej práce, pani prof. Ing. Danici Rosinovej, PhD., ktorá rovnako vníma danú problematiku ako aktuálny smer vývoja v oblasti priemyslu, ktorá ma podporovala pri štúdiu a bola vždy ochotná pomôcť a poradit.

Obsah

Úvod	1
1 Novodobé smerovanie informačno-komunikačných technológií v priemysle	4
1.1 Industry 4.0	5
1.1.1 Komponenty Industry 4.0.....	6
1.1.2 Princípy návrhu infraštruktúry postavenej na Industry 4.0	7
1.1.3 Možné využitia princípu Industry 4.0	8
1.2 Cloudové výpočty	10
1.3 Internet vecí.....	15
1.3.1 Vrstvy architektúry IoT	17
1.3.2 Komunikačné prostriedky IoT	19
1.3.3 Typy implementácie IoT	26
1.4 Digitálne dvojča	27
2 Umelá inteligencia a umelé neurónové siete	31
2.1 Princíp umelých neurónových sietí.....	33
2.2 Hlboké neurónové siete.....	36
2.3 Konvolučné neurónové siete.....	38
2.4 Rekurentné neurónové siete	40
2.5 Konfigurácia umelých neurónových sietí	42
3 Inteligentná diagnostika v priemysle.....	44
4 Ciele dizertačnej práce	53
5 Návrh systému monitorovania a riadenia mechatronických systémov	56
5.1 Požiadavky na výsledný systém.....	56
5.1.1 Funkcionálne požiadavky na výsledný systém	56
5.1.2 Nefunkcionálne požiadavky na výsledný systém.....	57
5.2 Koncept výsledného systému.....	57
5.2.1 Úroveň zariadenia	57
5.2.2 Úroveň clodu	58
5.2.3 Úroveň prepojenia	59
6 Implementácia systému monitorovania a riadenia mechatronických systémov.....	61

6.1	Mechatronické zariadenie	61
6.1.1	Vlastnosti a špecifikácia zariadení	62
6.1.2	Ovládanie zariadenia	65
6.1.3	Komunikácia zariadenia s cloudom	66
6.2	Inteligentné vyhodnocovanie stavu zariadenia	67
6.2.1	Dáta definujúce zariadenia	68
6.2.2	Architektúra umelej neurónovej siete	71
6.2.3	Princíp vyhodnocovania stavu zariadenia navrhnutým riešením inteligentnej diagnostiky	75
6.3	Google Cloud Platform (GCP).....	76
6.3.1	GCP Pub-Sub	78
6.3.2	GCP IoT Core.....	79
6.3.3	GCP BigQuery	81
6.3.4	GCP Storage – Bucket.....	82
6.3.5	GCP AI Platform.....	83
6.3.6	GCP Cloud functions	85
7	Testovanie navrhnutého riešenia a zhodnotenie výsledkov	87
8	Diskusia	95
	Záver	97
	Publikačná činnosť autora	100
	Zoznam použitej literatúry.....	102
	Prílohy	I
A.	Štruktúra elektronického nosiča	II

Zoznam obrázkov a tabuliek

Obr. 1 Priemyselné revolúcie	4
Obr. 2 Infraštruktúra podľa Industry 4.0	7
Obr. 3 Komunikácia clodu so zariadeniami	10
Obr. 4 Cloudové modely nasadenia.....	13
Obr. 5 „ <i>Internet vecí je v podstate jeden obrovský cloud.</i> “ [7].....	15
Obr. 6 Klasický IoT systém a jeho chod	17
Obr. 7 Vrstvy architektúry IoT	18
Obr. 8 AMQP komunikácia.....	20
Obr. 9 CoAP komunikácia.....	21
Obr. 10 DDS komunikácia	22
Obr. 11 MQTT komunikácia	23
Obr. 12 XMPP komunikácia	24
Obr. 13 Vývoj Digitálneho dvojčaťa	27
Obr. 14 Tri rozdelenia Digitálnych Dvojčiat podľa úrovne integrácie	29
Obr. 15 Strojové učenie podľa zaradenia	32
Obr. 16 jednoduchá neurónová siet'.....	33
Obr. 17 Porovnanie aktivačných funkcií sigmoid, tanh, ReLu, LReLu [20]	35
Obr. 18 Výkon NN vzhľadom k objemu dát a veľkosti jej architektúry [22]	37
Obr. 19 Plne prepojená hlboká neurónová siet'	37
Obr. 20 Konvolučná operácia.....	39
Obr. 21 Max pooling	40
Obr. 22 Grafické znázornenie výpočtu aktivačných funkcií g_1 a g_2 a ich vplyvu na výstup a časové kontinuum [24].....	42
Obr. 23 Predikcia zdravia motora pomocou dvoch ANN [25].....	44
Obr. 24 Použitie pred-trénovaného modelu pre diagnostiku zariadenia [26].....	45
Obr. 25 Kalibrácia digitálneho modelu na základe odozvy reálneho zariadenia [27].....	46
Obr. 26 ADCNN architektúra [28]	47
Obr. 27 TICNN architektúra [29]	48
Obr. 28 Klasifikačná CNN siet' pre vstupy o veľkosti 64x64 [31].....	48
Obr. 29 Architektúra GRU-NP-DAE [32].....	49

Obr. 30 Vanilla LSTM [33]	50
Obr. 31 Architektúra využívajúca zakódované vlastnosti z auto-enkóderov pri klasifikácii stavu zariadenia. [34].....	51
Obr. 32 Základný návrh inteligentnej diagnostiky a riadenia daného zariadenia	54
Obr. 33 Štruktúra prepojenia jednotlivých komponentov úrovni konceptu riešenia v prostredí Google Cloud Platform.....	59
Obr. 34 Popis zariadenia FloatShield [37].....	62
Obr. 35 Obvod zapojenia zariadenia FloatShield a Arduino vývojovej dosky [37].....	63
Obr. 36 Ukážka zariadenia MotoShield [37].....	64
Obr. 37 Obvod zapojenia zariadenia MotoShield a Arduino vývojovej dosky [37]	65
Obr. 38 Zariadenie FloatShield / MotoShield a jeho komunikácia s cloudom.....	66
Obr. 39 U-Net architektúra so vstupom 572x572 [39]	72
Obr. 40 xAxis U-Net	74
Obr. 41 Predikcia kĺzavým oknom	76
Obr. 42 Vytvorenie témy pre publikovanie	78
Obr. 43 Vytvorenie témy na odoberanie	79
Obr. 44 Vytvorenie IoT Core registra v GCP.....	80
Obr. 45 Registrovanie zariadenia v GCP.....	81
Obr. 46 Vytvorenie tabuľky v databáze BigQuery.....	82
Obr. 47 Vytvorenie bucketu na GCP	83
Obr. 48 Vytvorenie modelu	84
Obr. 49 Vytvorenie verzie modelu	85
Obr. 50 Vytvorenie udalosťou spúšťanej funkcie v prostredí GCP	86
Obr. 51 Priebeh chybovej funkcie naprieč 100 epochami pre prvých 10 rôznych nastavení	88
Obr. 52 Priebeh vývoja MCC metriky naprieč 100 epochami pre prvých 10 rôznych nastavení	88
Obr. 53 Chybová funkcia vybraného modelu po druhom kole hľadaní hyperparametrov ..	90
Obr. 54 MCC metrika vybraného modelu po druhom kole hľadaní hyperparametrov	90
Obr. 55 Matica zámeny pre FloatShield	92
Obr. 56 Matica zámeny pre MotoShield	94

Tabuľka 1 Koncepty Industry 4.0, vlastnosti, technológie a možné použitia [5].....	9
Tabuľka 2 Porovnanie základných vlastností IoT komunikačných protokolov [9]	24
Tabuľka 3 Sumár IoT protokolov [9]	25
Tabuľka 4 Komunikácia medzi ESP8266 a ATMega328 pre FloatShield a MotoShield ...	67
Tabuľka 5 Kompozícia datasetu pre diagnostiku zariadenia FloatShield	69
Tabuľka 6 Kompozícia datasetu pre diagnostiku zariadenia MotoShield	70
Tabuľka 7 Schéma tabuľky obsahujúca predikcie pre jednotlivé časové body.....	82
Tabuľka 8 Prvé spustenie hľadania hyperparametrov	87
Tabuľka 9 Druhé spustenie hľadania hyperparametrov	89
Tabuľka 10 Výsledky MCC evaluačných metrík získaných modelov na testovacích dátach	91
Tabuľka 11 Klasifikačné skóre diagnostického algoritmu pre FloatShield	91
Tabuľka 12 Klasifikačné skóre diagnostického algoritmu pre MotoShield	93

Zoznam skratiek a značiek

- AI – Artificial Intelligence, Umelá inteligencia
- AMQP – Advanced Message Queuing Protocol, komunikačný protokol
- ANN – Artificial Neural Networks, umelé neurónové siete
- API – Application Programming Interface, Rozhranie pre programovanie aplikácií
- Auto-ID – Automatic identification, Automatická identifikácia
- AV – Augmented virtuality, Rozšírená virtualita
- BDA – Big data analytics, Analýza big data
- BLE – Bluetooth Low Energy, Bluetooth s nízkou spotrebou energie
- CNN – Convolutional Neural Networks, konvolučné neurónové siete
- CoAP – Constrained Application Protocol, komunikačný protokol
- CPS – Cybernetic-Physical Systems, Kyberneticko-fyzikálne systémy
- CRM – Customer Relationship Management, Riadenie vzťahov so zákazníkmi
- DDS – Data Distribution Service, komunikačný protokol
- DEKG – Device electrocardiogram, elektrokardiogram zariadenia
- DNN – Deep Neural Networks, hlboké neurónové siete
- DSL – Digital Subscriber Line, Rodina technológií, ktoré poskytujú digitálny prenos údajov po drôtoch miestnej telefónnej siete
- DT – Digital Twin, Digitálne dvojča
- DTLS – Datagram Transport Layer Security, zabezpečovací protokol
- GCP – Google Cloud Platform
- GPRS – General Packet Radio Service, Univerzálna paketová rádiová služba
- HR – Human Resource, Ľudské zdroje
- HTTP – The Hypertext Transfer Protocol, Hypertextový prenosový protokol,
- IMS – Intelligent manufacturing system, Inteligentný systém výroby
- IaaS – Infrastructure as a Service, Infraštruktúra ako služba
- ID – Identifikátor
- IDaaS – Identity as a Service, Identita ako servis
- IIoT – Industrial Internet of Things, Priemyselný internet vecí
- IKT – Informačno-komunikačné technológie
- IoS – Internet od Services, Internet služieb

IoT – Internet of Things, Internet vecí

IP – Internet Protocol, Internetový protokol

ISO – International Organization for Standardization, Medzinárodná organizácia pre normalizáciu

IT – Information Technology, Informačné technológie

ITU – International Telecommunications Union, Medzinárodná telekomunikačná únia

JSON – JavaScript Object Notation, typ súboru

LAN – Local Area Network, Lokálna počítačová siet'

M2M – Machine to Machine, komunikácia stroj - stroj

M2M – Machine to Machine, komunikácia stroj - stroj

MCU – Microcontroller, Mikropočítač

MLP – Multilayer perceptron, viacvrstvový perceptrón

MPU – Microprocessor, Mikroprocesor

MQTT – Message Queuing Telemetry Transport, komunikačný protokol

MQTT-SN – Message Queuing Telemetry Transport for Sensor Networks, komunikačný protokol

NaaS – Network as a Service, Sieť ako servis

NN – Neural Networks, neurónové siete

OMG – Object Management Group, vývojár programovacieho jazyka

OSI – Open Systems Interconnection, Otvorené prepojenie systémov

PaaS – Platform as a Service, Platforma ako servis

PC – Personal Computer, Osobný počítač

PLM – Product Lifecycle Management, Systémy riadenia životného cyklu produktu

RNN – Recurrent Neural Networks, rekurentné neurónové siete

QoS – Quality of service, Kvalita služby

RAM – Random Access Memory, Pamäť s priamym prístupom

REST – Representational State Transfer, Reprezentačný stavový transfer

RFID – Radio frequency identification, Vysokofrekvenčná identifikácia

SaaS – Software as a service, Softvér ako servis

SASL – Simple Authentication and Security Layer, metóda na pridávanie alebo zlepšovanie overovania v protokoloch klient-server

SDK – Software development kit, Súbor nástrojov pre vývoj softvéru

SSAE – Stacked sparse auto encoder, skladaný riedky auto-enkóder

SSL – Secure Sockets Layer, Vrstva bezpečných socketov

STEP – Standard for the Exchange of Product Model Data, štandard pre výmenu dát modelu produktu

TCP – Transmission Control Protocol, protokol transportnej vrstvy

TLS – Transport Layer Security, protokoly, ktoré slúžia na šifrovanie dát

UDP – User Datagram Protocol, Používateľský datagramový protokol

UMTS – Universal Mobile Telecommunications System, jedna z technológií 3G

URL – Uniform Resource Locator, Jednotný vyhľadávač prostriedku, odkaz

USB – Universal Serial Bus, Univerzálna sériová zbernice

VLAN – Virtual Local Area Network, Virtuálna lokálna siet'

VPN – Virtual Private Network, Virtuálna privátna siet'

WAN – Wide Area Network, spojenie niekoľko sietí LAN

WebRTC – Web Real-Time Communication, komunikačný protokol

WIP – Work-in-progress, Práve vo výrobe

XML – eXtensible Markup Language, rozšíriteľný značkovací jazyk

Úvod

Vyspelé krajiny dnes rýchlo reagujú na celosvetovú výzvu potreby rozvoja a modernizácie priemyselných výrob. V mnohých európskych krajinách veľké spoločnosti vyvíjajú svoje vlastné metódy, prostriedky a stratégie na riešenie týchto naliehavých úloh. Súčasný stav v moderných digitalizovaných výrobných a technologických procesoch núti najmä malé a stredné podniky, aby sa prispôsobili týmto výzvam a budovali moderné digitalizované továrne spolupracujúce s veľkými spoločnosťami, najmä v strojárskom a automobilovom priemysle. V mnohých krajinách je koncepcia inteligentného priemyslu národnou iniciatívou založenou na najnovšom výskume uskutočnenom na univerzitách a vo firmách s cieľom transformovať a posilniť odvetvia pomocou metodológie Industry 4.0. Spojenie pokročilých technologických a výrobných procesov s informačnými a komunikačnými technológiami, automatizáciou, umelou inteligenciou, Big Data, Analytics a Cloudových technológií je hnacou silou súčasnej priemyselnej revolúcie.

Industry 4.0 je aktuálna celosvetová výzva, ktorá umožňuje interdisciplinárne využitie širokého spektra vedných a aplikačných oblastí tak, aby bolo možné synergicky integrovať najnovšie poznatky z metód a algoritmov automatického riadenia, elektroniky, informačno-komunikačných technológií (IKT) a umelej inteligencie do nových výrobkov, technológií, priemyselných procesov, zdravotníctva alebo služieb.

Kyberneticko-fyzikálne (CPS) systémy reprezentujú základnú platformu Priemyslu 4.0, podľa ktorej sa dnes budujú moderné digitalizované inteligentné fabriky schopné zefektívniť svoju činnosť na všetkých úrovniach riadenia, autonómnej výmeny a spracovania informácií medzi strojmi, snímačmi a riadiacimi systémami pre zabezpečenie rýchlej a intelligentnej činnosti riadenia. CPS ako základná platforma Industry 4.0 predpokladá vzájomnú spoluprácu a komunikáciu rozhraní človek – stroj, človek – človek, stroj – stroj, a aj komunikáciu v rámci logistických systémov a produktov. Na uvedený trend je nevyhnutne potrebné reagovať aj z pozície vývoja, výskumu, implementácie a realizácie metód a algoritmov riadenia procesov. Kyberneticko-fyzikálna platforma predstavuje také integrované komplexné systémy, ktoré s využitím metód a algoritmov intelligentného riadenia, IKT, inteligentných prvkov a systémov, založených na Internete vecí (IoT), dokážu dať výrobkom a technológiám novú a vylepšenú funkciu, nové konštrukčné vlastnosti a optimálnu súčinnosť dielčích procesov.

Ukazuje sa, že pre efektívne spracovanie dát môžeme dnes využiť clouдовé výpočty, zatiaľ čo kyberneticko-fyzikálne systémy ovplyvňujú fyzické procesy a ich výstupy sa menia na základe vstupov z fyzického sveta. Celý princíp konceptu Industry 4.0 je dnes založený najmä na dátach, IoT a IIoT. Základom modernej štruktúry výrob je digitalizácia procesov od úrovne senzorov a regulátorov až po vyššie úrovne pre optimalizáciu a plánovanie výrob. Každá z týchto úrovní má určitý vstup a výstup, ktorý sa spracováva ďalej a posielá na vyššie úrovne. Tu je možné hovoriť o interaktívnom prepojení senzorov, zariadení, strojov, komplexných riešení, ľudí, zdrojov, plánovania a výsledných produktov. Hranica medzi digitálnym a fyzickým svetom sa stráca, čo zlepšuje pochopenie celého životného cyklu priemyslu a jeho častí. Čím sa pochopenie týchto prepojení prehlbuje, tým narastá aj počet možností a príležitostí na vhodnejšie a rýchlejšie reagovanie na zmeny a získané skutočnosti o stave systému – fyzického zariadenia.

Predložená dizertačná práca je motivovaná technologickým rozvojom a novými požiadavkami automatizácie a digitalizácie procesov na základe inteligentných prístupov monitorovania a riadenia kybernetických/mechatronických zariadení. Rýchlosť napredovania technických výmoženosťí časom neustále rastie a tak je nutné pohotovo reagovať na vznikajúce zmeny tak, aby bola zabezpečená čo najvyššia efektivita prebiehajúcich procesov akými sú napríklad optimálne plánovanie, digitalizovaná výroba, alebo aj prediktívna údržba.

Cieľom dizertačnej práce je výskum, vývoj a nasadenie inteligentných postupov diagnostiky mechatronických zariadení v sieti IIoT v súlade s konceptom Industry 4.0 tak, aby boli zariadenia na základe výsledku analýzy obslužené a tak sa zamedzilo fatálnym následkom spojených so zlyhaním zariadenia.

Dizertačná práca využíva aktuálne výzvy moderných postupov modelovania a riadenia priemyselných procesov, ktoré sú dnes reprezentované kyberneticko-fyzikálnymi systémami. Jednou z najdôležitejších úloh pri zavádzaní metodológie konceptu Industry 4.0 do výrob je inteligentná diagnostika stavu zariadení použitím metód umelej inteligencie, čo predstavuje hlavné zámery predloženej dizertačnej práce.

V dizertačnej práci je navrhnutý, spracovaný a otestovaný koncept intelligentnej diagnostiky skupiny zariadení a ich riadenia na základe výsledkov diagnostiky v reálnom čase. Koncept je založený na metodológii Internetu vecí, kedy zariadenia komunikujú so vzdialou výpočtovou jednotkou. Táto výpočtová jednotka sa nachádza

v decentralizovanom priestore clodu, kde spracováva prichádzajúce informácie o chode zariadenia a pomocou umelej inteligencie sa rozhoduje o stave zariadenia. Výsledky diagnostiky zariadenia zapíše do databázy a na základe dosiahnutých výsledkov vyhodnotí, či sa zariadenie nachádza v kritickom stave.

Pôvodným a hlavným výsledkom dizertačnej práce je vývoj a implementácia inovatívnych prístupov inteligentnej diagnostiky a riadenia na základe monitorovania a návrhu unifikovanej intelligentnej diagnostiky kyberneticko-fyzikálnych zariadení na báze umelej inteligencie, ktoré sa opierajú o metódy hlbokého učenia, ktoré autorka dizertačnej práce rozšírila, modifikovala a pripravila na reálne použitie. Jedným z nových prínosov riešenia je aj zovšeobecnený koncept využívania získavania údajov o CPS zariadení. Ďalším prínosom práce je autorkou navrhnutý koncept spracovania získaných údajov tak, aby bolo možné realizovať diagnostiku stavu monitorovacieho zriadenia pomocou metód umelej inteligencie na diaľku - v decentralizovanom priestore clodu pre široké spektrum zariadení.

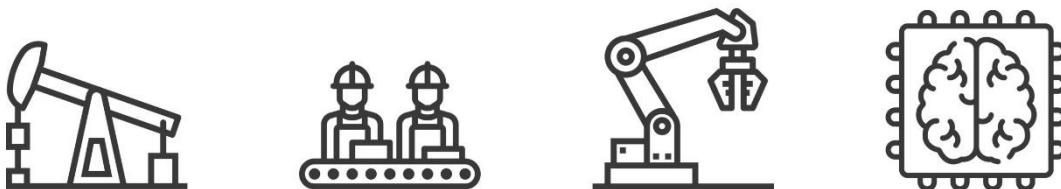
1 Novodobé smerovanie informačno-komunikačných technológií v priemysle

V prvej kapitole dizertačnej práce sa venujeme aktuálnym trendom v oblasti informačno-komunikačných technológií v priemysle. V prvej podkapitole sa venujeme konceptu a paradigme známej ako Industry 4.0. Venujeme sa jednotlivým komponentom, ktoré tento koncept tvoria, princípom návrhu infraštruktúry postavenej na koncepte Industry 4.0 a následne aj možným použitiam a aplikáciám tohto aktuálneho technologického smeru.

Technológie sa stali neoddeliteľou súčasťou ľudskej spoločnosti. Ich účelom je zvýšenie komfortu a zefektívnenie nášho každodenného života. Do značnej miery sme už v dnešnej dobe nimi ovplyvnení. V modernej spoločnosti je temer nemožné nájsť jej príslušníka bez spojenia s technológiami. Nemyslia sa tým iba výdobytky poslednej doby, pretože ich súčasťou sú napríklad aj dávne parné lokomotívy, či aj veci každodennej spotreby ako napríklad rýchlovarná kanvica, práčka či obyčajná lampa.

Postupom času sa technológie neustále zlepšovali a stále napredujú. Ako ľudstvo sme si prešli niekoľkými priemyselnými revolúciami a práve Industry 4.0, na ktorej prahu stojíme, je ďalším krokom vo vývoji vpred.

Prvá priemyselná revolúcia je väčšine ľudí známa, nakoľko bola pre ľudstvo samotné prelomová, no nie je na škodu spomenút', čo vlastne samotnej štvrtej priemyselnej revolúcii predchádzalo. Prehľad predchádzajúcich priemyselných revolúcií je na



1. revolúcia

mechanizácia
energia vody
energia par

2. revolúcia

masová výroba
pásová výroba
elektrina

3. revolúcia

digitalizácia
automatizácia

4. revolúcia

kyberneticko-fyzické
systémy

Obr. 1 Priemyselné revolúcie

Prvým a to zásadným krokom, ktorý radikálne zmenil životy ľudí, bola samotná mechanizácia. Za použitia vody a pary bolo možné rozhýbať aj niekoľko tonové stroje. Niečo, čo bolo predtým nemysliteľné vykonat'. Týmto bol položený základný kameň výrobných odvetví pre dnešný svet, v ktorom sa nachádzame. Zmeny bolo možné vidieť v každom smere, no predovšetkým v poľnohospodárstve a výrobe textilu a pojmom „továreň“ sa začal rozširovať do povedomia ľudí.

Za druhú priemyselnú revolúciu sa považuje obdobie, kedy sa začala masová produkcia za použitia elektrickej energie. Pásová výroba, ktorá je dodnes známa uzrela svetlo sveta. Elektrická energia priniesla ale aj množstvo ďalších benefitov, vďaka ktorým produkcia prudko vzrástla.

Ďalší, tretí, krok v priemysle prišiel v momente, kedy sa začala uplatňovať elektronika a informačné technológie pre automatizáciu výroby. Častokrát je možné počuť pojmom „Digitálna revolúcia“, ktorá označuje práve toto obdobie. Jej hlavnou podstatou je prechod z mechanických a analógových prístrojov na digitálne.

Štvrtá priemyselná revolúcia prináša automatizáciu výrobných procesov na novú úroveň zavedením prispôsobených a flexibilných výrobných technológií. To znamená, že stroje budú pracovať nezávisle alebo spolupracovať s ľuďmi pri vytváraní zákaznícky orientovaného produkčného prostredia, ktoré neustále pracuje na vlastnom udržiavaní. Stroj sa stáva nezávislým subjektom, ktorý dokáže dátá zhromažďovať, analyzovať a využívať. To sa dá dosiahnuť zavedením učiacich sa systémov: seba-optimalizácie, seba-poznávania a seba-prispôsobovania do priemyslu. Takýmto spôsobom výrobcovia skôr so strojmi spolupracujú, než ich priamo ovládajú.

1.1 Industry 4.0

V posledných niekoľkých desaťročiach sa objavila štvrtá priemyselná revolúcia známa ako Priemysel 4.0 (Industry 4.0). Priemysel 4.0 kladie dôraz na digitálnu technológiu z posledných desaťročí a posúva ju na úplne nový level pomocou prepojenia medzi IoT, prístupu k dátam v reálnom čase a využitiu prístupu kyberneticko-fyzikálnych systémov. Industry 4.0 ponúka v porovnaní s predošlými priemyselnými konceptami komplexnejší, prepojený a holistický prístup k výrobe. Spája fyzické s digitálnym a umožňuje lepšiu spoluprácu a prístup medzi oddeleniami, partnermi, dodávateľmi, výrobkami a ľuďmi. Postupnou digitalizáciou sa výroba presúva k tzv. Smart Factory. Ako výsledok takého

prístupu sa očakáva postupné prepojenie medzi každou jednotkou výroby od najmenšieho článku až po komplex zdrojov a udalostí. Takéto prepojenie si sľubuje zabezpečenie výrazne vyššieho zefektívnenia práce na základe využitia nesmierneho množstva informácií, ktoré doteraz nebolo možné včas zachytiť a spracovať, podstatne rýchlejšie a vhodnejšie tak možno reagovať na jednotlivé podnety. Práve takéto obsiahlejšie monitorovanie jednotlivých strojov v rámci výroby napomáha rýchlejšiemu a efektívnejšiemu spracovaniu zmien, ktoré nastanú.

1.1.1 Komponenty Industry 4.0

Za hlavné komponenty Smart Factory, známej aj ako Inteligentná továreň, pre zabezpečenie aplikácie Industry 4.0 sú predovšetkým:

- **Kyberneticko-fyzikálne systémy (CPS)** – sú integrácie výpočtov a fyzických procesov. Vstavané počítače a siete sledujú a riadia fyzické procesy, zvyčajne so spätnou väzbou, kde fyzické procesy ovplyvňujú výpočty a naopak [1].
- **Internet vecí (IoT)** – môže byť široko definovaný ako celosvetová sieťová infraštruktúra, ktorá spája jedinečne identifikované fyzické a virtuálne objekty, veci a zariadenia prostredníctvom inteligentných objektov, komunikačných a akčných schopností [2].
- **Internet služieb (IoS)** - má za cieľ vytvoriť akýsi obal všetkých zariadení pripojených k službe, ktorý slúži pre ich združenie a zber dát ako jedného zariadenia. Je to pomyselná vstupná brána zákazníka k výrobcovi.
- **Big Data** – Samotný termín nepopisuje určitú veľkosť dát, ale práve prácu s nimi pri prediktívnych analýzach a iných pokročilých metódach extrakcie dát pre ich bližší rozbor a ďalšiu prácu s nimi [3].
- **Cloudové výpočty** – sa týkajú aplikácií dodávaných ako služby cez internet a hardvérového a systémového softvéru v dátových centrách, ktoré poskytujú tieto služby [4]. Ponuka aplikácií sa pohybuje od kancelárskych aplikácií, cez systémy pre distribuované výpočty [3].
- **Systémy riadenia životného cyklu produktu** – Product Lifecycle Management (PLM) má za úlohu riadiť a spravovať detailné informácie o konštrukčnom riešení produktu, jeho vlastnostiach, používaní i spôsobe výroby. Umožňuje tieto dátá integrovať spolu s procesmi, obchodnými systémami, dodávateľmi, zdrojmi atď. [3].

- **Digitálna výroba** – je postup, ktorý je založený na simuláciách, vizualizáciách, analýzach a ďalších nástrojoch pre tvorbu samotného produktu. Digitálna výroba sa považuje za hlavný bod pri integrácii PLM rôznorodými zariadeniami a aplikáciami, zabezpečujúcimi výrobu a kontrolu dielov [3].
- **Digitálne dvojča** – Podstata digitálneho dvojča leží v premietnutí fyzického zariadenia do virtuálneho sveta. Táto transformácia môže byť použitá pre rôznorodé účely, ako napríklad simulácia chodu zariadenia či vizualizácia vzdialeneho zariadenia [3].

1.1.2 Princípy návrhu infraštruktúry postavenej na Industry 4.0

Kombináciou vyššie spomenutých komponentov sa dostávame k samotným princípm návrhu infraštruktúry Industry 4.0. Jednotlivé princípy návrhu infraštruktúry Industry 4.0 sú interoperabilita, virtualizácia, decentralizácia, reálny čas, orientácia na služby a modularita.



Obr. 2 Infraštruktúra podľa Industry 4.0

- **Interoperabilita** – Objekty, stroje a ľudia musia byť schopní komunikovať prostredníctvom internetu vecí a internetu ľudí. To je najdôležitejší princíp, ktorý skutočne robí továreň inteligentnou.
- **Virtualizácia** – CPS musia byť schopné simulať a vytvoriť virtuálnu kópiu skutočného sveta. CPS musia byť tiež schopné sledovať objekty, ktoré existujú v okolitom prostredí. Jednoducho povedané, musí existovať virtuálna kópia všetkého.
- **Decentralizácia** – Schopnosť CPS pracovať nezávisle. To dáva priestor pre prispôsobené produkty a riešenie problémov. To tiež vytvára flexibilnejšie prostredie pre výrobu. V prípade zlyhania alebo konfliktných cieľov je problém delegovaný na vyššiu úroveň. Avšak aj pri zavádzaní takýchto technológií zostáva potreba zabezpečenia kvality nevyhnutnosťou celého procesu.

- **Reálny čas** – Inteligentná továreň musí byť schopná zhromažďovať údaje v reálnom čase, ukladať alebo analyzovať a robiť rozhodnutia podľa nových zistení. Toto nie je obmedzené iba na prieskum trhu, ale aj na interné procesy, ako napríklad zlyhanie stroja vo výrobnej linke. Inteligentné objekty musia byť schopné identifikovať chybu a znova delegovať úlohy na iné prevádzkové stroje. To tiež výrazne prispieva k flexibilite a optimalizácii výroby.
- **Orientácia na služby** – Produkcia musí byť orientovaná na zákazníka. Ľudia a inteligentné objekty / zariadenia musia byť schopné efektívne sa pripojiť cez internet služieb a vytvoriť produkty na základe špecifikácií zákazníka. Tu sa stáva dôležitý internet služieb.
- **Modularita** – Na dynamickom trhu je nevyhnutná schopnosť Inteligentnej továrne prispôsobiť sa novým požiadavkám trhu. V typickom prípade pravdepodobne bude trvať týždeň, kým priemerná spoločnosť vykoná štúdie na trhu a zodpovedajúco zmení svoju výrobu. Na druhej strane sa inteligentné továrne musia rýchlo a hladko prispôsobiť sezónnym zmenám a trendom na trhu úpravou svojich produktov a výrobných procesov. Jednotlivé súčasti by mali byť základom pre technologický model Industry 4.0, ktorý je možné aplikovať do ktoréhokoľvek výrobného procesu.

1.1.3 Možné využitia princípu Industry 4.0

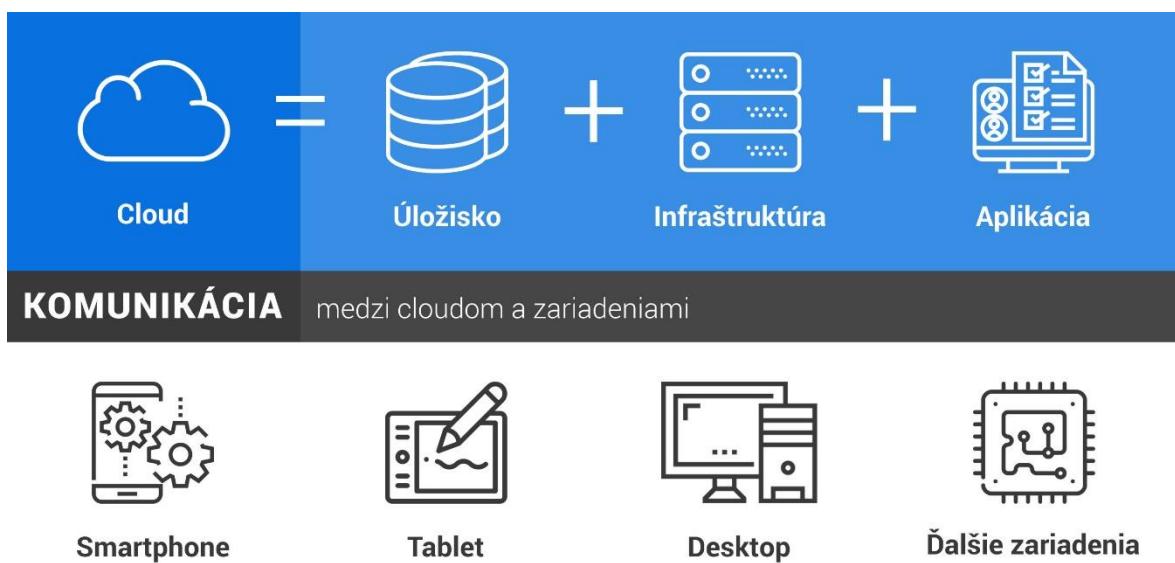
V nasledujúcej tabuľke môžeme vidieť určité koncepty Industry 4.0, ich stručný opis, podporované technológie a aplikačné prostredia, v ktorých sa dajú uplatniť. Koncepty Industry 4.0 je možné zaradiť do troch hlavných kategórií: inteligentná výroba, IoT výroba a clouдовá výroba. Tieto koncepty sú pomenované podľa najviac využívaných technológiách a ich zameraní, avšak princíp Industry 4.0 stojí na prepojení jednotlivých komponentov a tak aj tieto koncepty využívajú viacero podporovaných technológií.

Tabuľka 1 Koncepty Industry 4.0, vlastnosti, technológie a možné použitia [5]

Koncept	Hlavné charakteristiky	Podporované technológie	Možné použitia
Inteligentná výroba	Inteligentné rozhodovanie založené na AI	Big data spracovanie	Inteligentný výrobný systém so zakomponovaním štandardu ISO STEP
	Pokročilá automobilová výroba	Pokročilá robotika	Testovanie životného cyklu produktu umožňujúce inteligentnú výrobu
	Adaptívne a flexibilné výrobné systémy	Priemyselné služby pripojenia Snímače poslednej generácie	Agentovo-založené IMS Inteligentné systémy plánovania a riadenia výroby
IoT výroba	Inteligentný výrobný systém založený na technológii Auto-ID	IoT	RFID založený systém riadenia zdrojov
	Zhromažďovanie údajov v reálnom čase	Bezdrôtová výroba	Inteligentný konštrukčný systém s podporou IoT
	Viditeľnosť a sledovateľnosť výrobných procesov v reálnom čase	BDA	Systém riadenia zásob WIP založený na technológii RFID
Cloud výroba	Rozhodovanie v reálnom čase vo výrobe	Cloudové výpočty	Systém plánovania produkcie v reálnom čase s RFID
	Distribúcia a zdieľanie služieb v oblasti výroby	Cloudové výpočty	Dátová vizualizácia produkcie cloudovej výroby
	Inteligentné riadenie schopností	IoT	Výber kompozície QoS služby v cloudovom systéme výroby
Spracovanie manažmentu clouдовých služieb	Spracovanie manažmentu clouдовých služieb	Virtualizačná metóda	Inteligentná cloudová výroba s použitím IoT
		Technológia orientovaná na služby	Sémantický webovo založený framework v cloudovej výrobe

1.2 Cloudové výpočty

Termín Cloud označuje siet' alebo internet. Inými slovami, môžeme povedať, že Cloud je niečo, čo je prítomné na diaľku. Služba Cloud môže poskytovať služby prostredníctvom verejných a súkromných sietí, t. j. WAN, LAN alebo VPN. Aplikácie ako e-mail, webové konferencie, riadenie vzťahov so zákazníkmi (CRM) sa vykonávajú v cloude.



Obr. 3 Komunikácia cloudu so zariadeniami

Cloudové výpočty sa týkajú manipulácie, konfigurácie a prístupu k hardvérovým a softvérovým zdrojom na diaľku. Ponúka on-line ukladanie dát, infraštruktúru a aplikácie. Cloudové výpočty ponúkajú nezávislosť platformy, pretože softvér sa nemusí nainštalovať lokálne na PC a preto vytvárajú mobilné a spolupracujúce biznis aplikácie.

Cloudové výpočty sú modelom, ktorý umožňuje všadeprítomný, pohodlný prístup k sieti na požiadanie k zdieľanému fondu konfigurovateľných výpočtových zdrojov (napr. Sietí, serverov, úložných zariadení, aplikácií a služieb), ktoré možno rýchlo poskytnúť a uvoľniť s minimálnym úsilím riadenia interakcie s poskytovateľom služieb. Takýto cloudový model pozostáva z esenciálnych charakteristík, servisných modelov a modelov nasadenia.

Esenciálne charakteristiky [6]:

- **Samoobslužná služba na požiadanie** – Spotrebiteľ si vie prispôsobiť výpočtové možnosti, napríklad čas servera a sietové ukladanie, podľa potreby automaticky bez požadovania interakcie s každým poskytovateľom služieb.
- **Široký prístup k sieti** – Prístup je zabezpečený prostredníctvom štandardných mechanizmov, ktoré podporujú používanie heterogénnych klientskych platform (napr. mobilov, tabletov, notebookov a pracovných staníc).
- **Združovanie zdrojov** – Výpočtové zdroje poskytovateľa sa združujú, aby slúžili viacerým spotrebiteľom s využitím modelu pre viacerých zákazníkov, s dynamickým priradením a priradením rôznych fyzických a virtuálnych zdrojov podľa spotrebiteľského dopytu. Zákazník nemá vo všeobecnosti žiadnu kontrolu alebo vedomosti o presnom umiestnení poskytnutých zdrojov, ale môže určiť polohu na vyšej úrovni abstrakcie (napr. krajinu, štátu alebo dátového centra). K zdrojom patrí úložisko, spracovanie, pamäť a šírka pásma siete.
- **Rýchla elasticita** – Možnosti môžu byť flexibilne zabezpečené a uvoľnené, v niektorých prípadoch sa môžu automaticky, a rýchlo meniť v súlade s dopytom. Spotrebiteľovi sa kapacity, ktoré sú k dispozícii na poskytovanie služieb, často java ako neobmedzené a môžu byť kedykoľvek zmenené.
- **Nameraná služba** – Cloudové systémy automaticky riadia a optimalizujú využívanie zdrojov tým, že využívajú meranie vytáženia. Využívanie zdrojov je možné monitorovať, kontrolovať a posielat správy o stave, čím poskytuje transparentnosť poskytovateľovi a spotrebiteľovi využívanej služby.

Servisné cloudové modely:

- **IaaS** – Infraštruktúra ako servis poskytuje prístup k základným zdrojom, ako sú fyzické počítače, virtuálne počítače, virtuálne úložiská, ukladanie diskov na virtuálnom počítači, virtuálna lokálna siet (VLAN), vyrovňávače načítavania, IP adresy alebo softvérové balíky. Všetky uvedené zdroje sú k dispozícii konečnému používateľovi prostredníctvom virtualizácie serverov. Okrem toho zákazníci pristupujú k týmto zdrojom, akoby ich vlastnili.
- **PaaS** – Platforma ako servis ponúka bežné prostredie pre aplikácie. Ponúka tiež nástroje vývoja a nasadenia potrebné na vývoj aplikácií. PaaS má funkciu nástrojov, ktoré

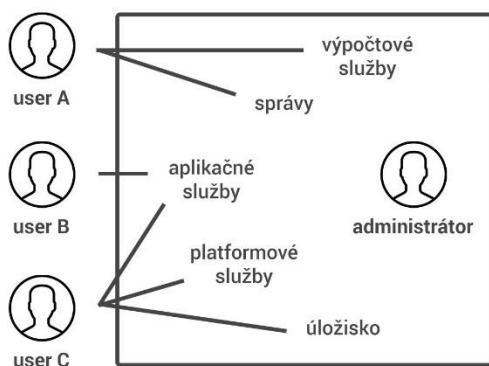
umožňujú vývojárom vytvárať webové aplikácie. Aplikácia Engine spoločnosti Google a Force.com sú príkladmi predajcov spoločnosti PaaS. Vývojár sa môže prihlásiť na tieto webové stránky a použiť vstavané rozhranie API na vytváranie webových aplikácií. Negatívom používania PaaS je to, že vývojár sa upíše konkrétnemu predajcovi. Napríklad aplikácia napísaná v Pythone v porovnaní s API spoločnosti Google a pomocou aplikácie App Engine spoločnosti Google pravdepodobne bude fungovať iba v tomto prostredí.

- **SaaS** – Model „softvér ako servis“ umožňuje poskytovať softvérovú aplikáciu ako službu koncovým používateľom. Ide o softvér, ktorý je nasadený na hostiteľskej službe a je prístupný cez internet. Aplikácie SaaS môžu byť napríklad systém fakturácie, aplikácie na riadenie vzťahov so zákazníkmi (CRM), aplikácie help desk, rozvoj ľudských zdrojov (HR). Niektoré aplikácie SaaS nie sú prispôsobiteľné, ako napríklad balík Microsoft Office Suite. Služba SaaS nám však poskytuje programovacie rozhranie API, ktoré umožňuje vývojárom vyvinúť prispôsobenú aplikáciu.
- **IDaaS** – Identita ako servis ponúka správu informácií o identite ako digitálnu entitu. Táto identita sa môže použiť počas elektronických transakcií. Takáto technika rieši problémy v situáciach ako je napríklad tá nasledovná: Zamestnanci v spoločnosti vyžadujú prihlásenie do systému na vykonávanie rôznych úloh. Tieto systémy môžu byť založené na lokálnom serveri alebo na cloude. Nasledujú problémy, ktorým môže zamestnanec čeliť, ako napríklad zapamätanie si rôznych kombinácií používateľského mena a hesla pre prístup k viacerým serverom, či ak zamestnanec opustí spoločnosť, je povinný zabezpečiť, aby bol každý účet tohto používateľa zakázaný. To zvyšuje pracovné zaťaženie pracovníkov IT.
- **NaaS** – Sieť ako servis umožňuje prístup k sieťovej infraštruktúre priamo a bezpečne. NaaS umožňuje nasadenie vlastných smerovacích protokolov. NaaS využíva virtualizovanú sieťovú infraštruktúru na poskytovanie sieťových služieb zákazníkovi. Je zodpovednosťou poskytovateľa NaaS udržiavať a spravovať sieťové zdroje. Mat' poskytovateľa, ktorý pracuje pre zákazníka a znižuje pracovné zaťaženie zákazníka. Okrem toho NaaS spoločnosť ponúka sieť ako nástroj. NaaS je tiež založená na modeli s platbou za použitie. Na použitie modelu NaaS je zákazník povinný prihlásiť sa na webový portál, kde môže získať on-line rozhranie API. Tu môže zákazník prispôsobiť trasu.

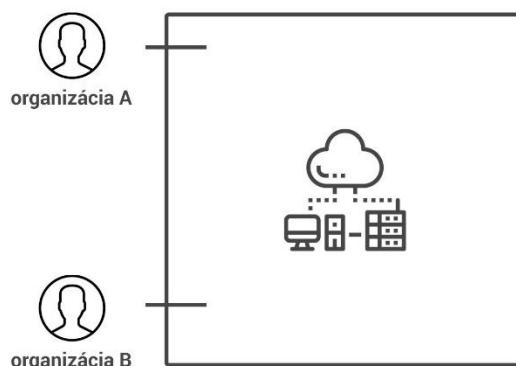
Cloudové modely nasadenia:

CLOUDOVÉ MODELY NASADENIA

verejný



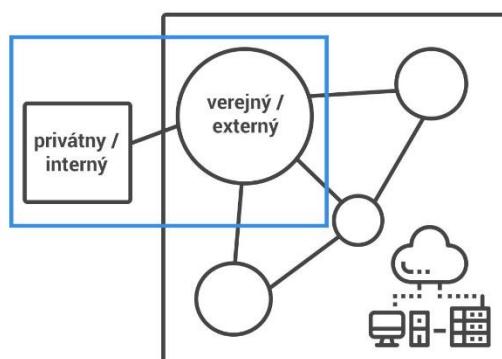
spoločenský



súkromný



hybridný



Obr. 4 Cloudové modely nasadenia

- **Verejný** – umožňuje, aby boli systémy a služby ľahko dostupné širokej verejnosti. IT giganti ako Google, Amazon a Microsoft ponúkajú cloudové služby cez internet. Výhodami takého clodu sú: efektivita výdavkov, spoločnosť, flexibilita, nezávislosť od lokality, vysoká škálovateľnosť. Tento model avšak prináša aj nevýhody ako napríklad nízke zabezpečenie a sú menej prispôsobiteľné.
- **Spoločenský** – umožňuje systém a služby aby boli prístupné určitej skupine organizácií. Zdieľa infraštruktúru medzi viacerými organizáciami z konkrétnej komunity. To môže byť interne spravované organizáciami alebo tretou stranou. Výhodou tohto modelu sú efektivita výdavkov, zdieľanie medzi organizáciami a lepšie zabezpečenie ako je to

u verejného servisného modelu. Jeho nevýhodou je uloženie dát na jednom mieste a preto si človek musí uvedomiť, že môžu byť prístupné aj ďalším organizáciám.

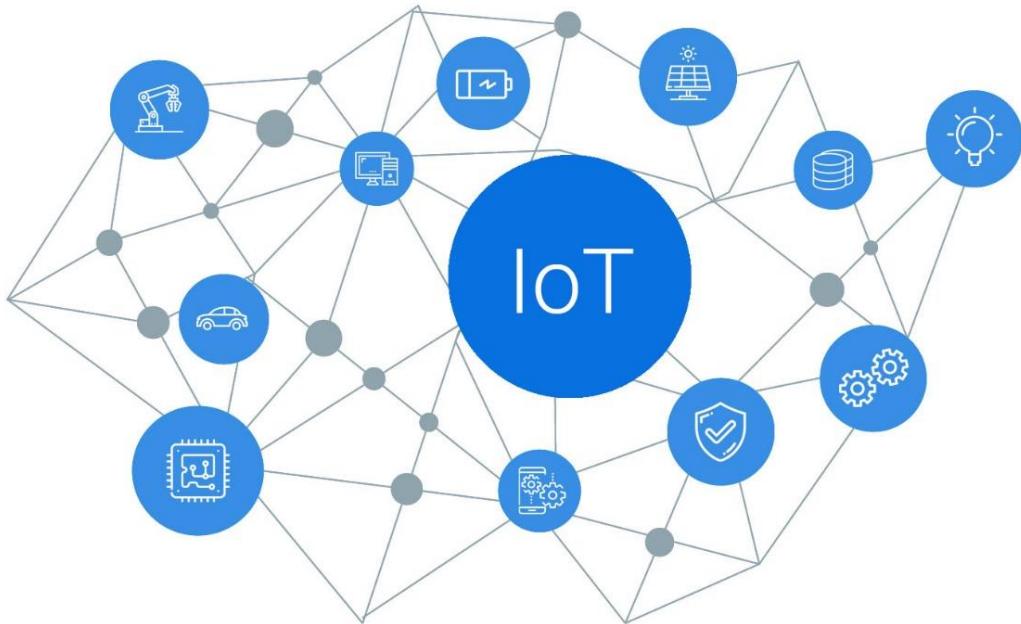
- **Súkromný** – umožňuje prístup k systémom a službám v rámci organizácie. Tento model funguje iba v rámci jednej organizácie. Môže sa však interne spravovať organizáciou alebo treťou stranou. Kladmi sú: vysoká miera zabezpečenia a súkromia, väčšia kontrola nad službami, väčšia efektivita. Medzi nevýhody sa radia: obmedzenie dosahu, vysoká cena, limitovaná škálovateľnosť v rámci fyzických zdrojov a samotná údržba.
- **Hybridný** – je zmes verejného a súkromného clodu. Nekritické aktivity sa vykonávajú pomocou verejného clodu, zatiaľ čo kritické aktivity sa vykonávajú pomocou súkromného clodu. Prednostami takéhoto riešenia sú: škálovateľnosť, flexibilita, cenová efektivita a zabezpečenie. K slabinám patria: problémy so sieťou vzhľadom k jej komplexnosti, súlad bezpečnostných opatrení clodu a samotnej organizácii a závislosť na internej IT infraštruktúre, preto je potrebné zabezpečiť redundanciu v dátových centrach.

Možnosti využitia:

- **Škálovateľné využitie** – Cloudové výpočty ponúkajú škálovateľné zdroje prostredníctvom rôznych predplatných modelov. To znamená, človek platí len za to, čo využíva. To pomáha pri riadení špičiek v nárokoch bez potreby trvale investovať do počítačového hardvéru. Napríklad **Netflix** využíva tento potenciál vo svoj prospech. Vďaka svojej streamingovej službe na požiadanie čelí veľkým nárastom zaťaženia servera v špičkových časoch. Pohyb migrácie z interných dátových centier na cloud umožnil spoločnosti výrazne rozšíriť svoju zákaznícku základňu bez toho, aby museli investovať do inštalácie a údržby nákladnej infraštruktúry.
- **Chatboty** – Rozšírený výpočtový výkon a kapacita clodu nám umožňuje uchovávať informácie o používateľských preferenciách. Toto je možné použiť na poskytovanie prispôsobených riešení, správ a produktov na základe správania a preferencií používateľov. **Siri, Alexa a Asistent Google** – všetci sú inteligentné roboty založené na technológii cloud. Tieto chatboty využívajú výpočtové možnosti clodu na poskytovanie personalizovaných skúseností so zákazníkmi relevantných pre kontext.
- **Komunikácia** – Cloud umožňuje používateľom využívať prístup založený na sieti na komunikačné nástroje ako e-maily a kalendáre. Väčšina aplikácií na zasielanie správ a

volaní, ako napríklad **Skype** a **WhatsApp**, je tiež založená na cloubovej infraštrukture. Všetky správy a informácie sú uložené na hardvéri poskytovateľa služieb a nie na osobnom zariadení. To umožňuje prístup k informáciám odkiaľkoľvek cez internet.

1.3 Internet vecí

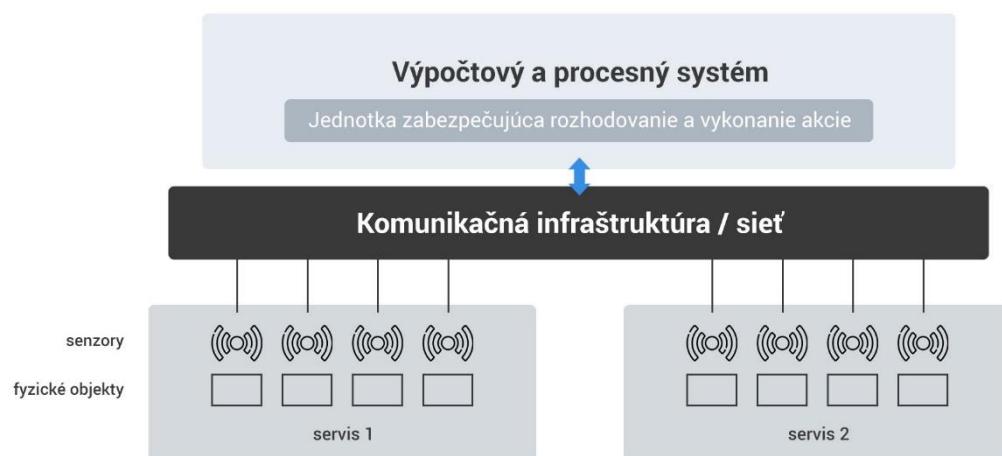


Obr. 5 „Internet vecí je v podstate jeden obrovský cloud.“ [7]

Internet vecí sa dá považovať vo svojej podstate za smer a ideu prepojenia fyzických objektov k internetu a prepojeniu medzi nimi samotnými pomocou bezdrôtových technológií. Tieto objekty zahŕňajú všetko od kávovaru až po výrobné stroje. Podstatou takéhoto prepojenia obrovského množstva rôznorodých objektov, ktoré sú rozdielne vo svojej funkcionalite, výrobcovi či dizajne, je zabezpečiť bezproblémovú a koherentnú skúsenosť pre konzumenta, či už sa jedná o bežného spotrebiteľa alebo o výrobnú halu či korporáciu.

Internet vecí je obrovská sieť pozostávajúca zo sietí zariadení a počítačov sériou sprostredkovateľských technológií, v rámci ktorých môžu pôsobiť početné technológie ako RFID, čiarové kódy, drôtové a bezdrôtové pripojenia. International Telecommunications Union (ITU) definovala Internet vecí ako štyri rozmery vecí [2]:

- **Označovanie vecí** – Vysledovateľnosť a adresovateľnosť RFID značiek v reálnom čase je tým, čo ich stavia do popredia, pokial' ide o víziu internetu vecí. RFID získavajú od podnikateľskej komunity silnú podporu z dôvodu ich splatnosti, nízkych nákladov a nízkej spotreby energie.
- **Cítiace veci (Feeling things)** – Snímače slúžia ako primárne zariadenia na zhromažďovanie údajov z prostredia. Potrebné údaje sa poskytujú prostredníctvom komunikácie medzi fyzickým a informačným svetom. Vďaka nedávnym pokrokom v oblasti technológií zariadenia spotrebujú menej energie s nízkymi nákladmi a vysokou účinnosťou.
- **Zmenšujúce sa veci** – Miniaturizácia a nano-technológia vyvolali schopnosť menších vecí komunikovať a spájať sa v rámci "vecí" alebo "inteligentných zariadení". Jasnu výhodou je zlepšenie kvality života. Napríklad aplikácia nano-senzorov na monitorovanie kvality vody za znížených nákladov alebo nano-membrány na pomoc pri čistení odpadových vôd. V oblasti zdravotnej starostlivosti sa jej aplikácia prejavuje v diagnostike a liečbe ochorení, vrátane diagnostiky HIV a AIDS a nano-liečivá pre iné choroby.
- **Mysliace veci** – Vytvorenie inteligentných zariadení pomocou dodania rôznorodých senzorov viedlo k dôležitej zmene pohľadu ako sa pripojiť k internetu. Vďaka senzorom vedia domáce elektrické zariadenia analyzovať údaje, ktoré im senzory poskytujú, a na základe tejto analýzy vhodne reagovať na zmeny. Napríklad chladničky, ktoré dokážu odhadnúť množstvo rôznych predmetov a čerstvosť tovaru podliehajúceho skaze. Integrované inteligentné snímače môžu poskytnúť prostriedky na komunikáciu s používateľmi odoslaním upozornenia prostredníctvom pripojenia na internet. Pripojenie môže byť primárne bezdrôtová alebo iná dostupná komunikácia, ako napríklad DSL, GPRS, WiFi, LAN a 3G. Nielen komunikovať, inteligentné veci musia byť schopné taktiež spracovávať informácie, konfigurovať vlastné nastavenia, vykonávať sebaobsluhu, opravovať sa, robiť nezávislé rozhodnutia alebo dokonca hrať aktívnu úlohu vo vlastnom mene, čo zmení spôsob komunikácie informácií človek-človek, človek-vec a vec-vec.



Obr. 6 Klasický IoT systém a jeho chod

Obr. 6 zobrazuje základný IoT systém implementujúci rôzne typy aplikácií alebo služieb. Fyzické objekty sa spájajú a komunikujú s inými objektami, ktoré implementujú rovnaký typ služby. Základný zjednodušený princíp funkcionality internetu vecí možno opísť ako:

Objektové snímanie, identifikácia a komunikácia informácií špecifických pre daný objekt. Informácie sú napríklad snímané údaje o teplote, orientácii, pohybe, vibráciách, zrýchlení, vlhkosti, chemických zmenách vo vzduchu atď. v závislosti od typu snímačov. Kombinácia rôznych senzorov môže byť použitá pre návrh inteligentných služieb.

1. Spustenie akcie. Informácie o prijatých objektoch spracováva inteligentné zariadenie alebo systém, ktorý potom určí automatickú akciu, ktorá sa má vyvolávať.
2. Intelligentné zariadenie / systém poskytuje bohaté služby a obsahuje mechanizmus na poskytnutie spätej väzby správcovi o aktuálnom stave systému a výsledkoch vyvolaných akcií.

1.3.1 Vrstvy architektúry IoT

Rozvoj internetu vecí závisí od technologického pokroku a návrhu rôznych nových aplikácií a obchodných modelov, preto bola základná architektúra IoT sformulovaná do 5 vrstiev: vrstva vnímania, siet'ová vrstva, sprostredkovateľská vrstva, aplikačná vrstva a biznis vrstva.



Obr. 7 Vrstvy architektúry IoT

Vrstva vnímania – je tiež známa ako "Vrstva zariadenia". Skladá sa z fyzických objektov a snímačov. Senzory môžu byť RFID, 2D-čiarový kód alebo infračervený senzor v závislosti od metódy identifikácie objektov. Táto vrstva sa v podstate zaobrá identifikáciou a zhromažďovaním špecifických informácií o objektoch pomocou senzorových zariadení. V závislosti od typu snímačov môžu byť informácie o polohe, teplote, orientácii, pohybe, vibráciách, zrýchlení, vlhkosti, chemických zmenách vo vzduchu atď. Zozbierané informácie sú potom odovzdané do sietovej vrstvy pre bezpečný prenos na spracovanie informácií systémom.

Sieťová vrstva – sa môže tiež nazývať "Prenosová vrstva". Táto vrstva bezpečne prenáša informácie zo senzorových zariadení do systému na spracovanie informácií. Prevodové médium môže byť drôtové alebo bezdrôtové a technológia môže byť 3G, UMTS, Wi-Fi, Bluetooth, infračervené, ZigBee atď. v závislosti od senzorových zariadení. Preto sietová vrstva prenáša informácie z vrstvy vnímania na sprostredkovateľskú vrstvu.

Sprostredkovateľská vrstva – Zariadenia cez IoT implementujú rôzne typy služieb. Každé zariadenie sa pripája a komunikuje iba s tými zariadeniami, ktoré implementujú rovnaký typ služby. Táto vrstva je zodpovedná za správu služieb a má spojenie s databázou. Prijíma informácie zo sietovej vrstvy a ukladá ich do databázy. Vykonáva spracovanie informácií a všadeprítomné výpočty a prijíma automatické rozhodnutie založené na výsledkoch.

Aplikačná vrstva – Táto vrstva poskytuje globálnu správu aplikácie na základe informácií o objektoch spracovaných v sprostredkovateľskej vrstve. Aplikácie implementované v rámci Internetu vecí môžu byť inteligentné zdravie, inteligentné polnohospodárstvo, inteligentné domácnosti, inteligentné mesto, inteligentná doprava atď.

Biznis vrstva – Táto vrstva je zodpovedná za riadenie celkového systému internetu vecí vrátane aplikácií a služieb. Vytvára obchodné modely, grafy, vývojové diagramy atď. založené na údajoch získaných z aplikačnej vrstvy. Skutočný úspech technológie Internetu vecí závisí aj od dobrých obchodných modelov. Na základe analýzy výsledkov táto vrstva pomôže určiť budúce akcie a obchodné stratégie [8].

V kontexte bežného spotrebiteľa IoT technológia môže predstavovať automatizáciu každodenných úloh. Napríklad senzory zaznamenajú zobudenie spotrebiteľa v predpokladanom čase, svetlá v miestnosti, kde sa užívateľ nachádza sa začnú postupne rozsvecovať, na budíku začne hrať rádio a v kuchyni začne ohrievať voda na kávu či čaj. IoT avšak nie je iba technológia pre každodenný život, ale aj pre biznis. Túto technológiu a dá využiť napríklad pri monitorovaní a automatickom spravovaní výrobného procesu, napr. emulzie keramiky. Pomocou senzorov je možné zaznamenať vlastnosti zmesi v určitých momentoch a časoch procesu a na základe toho analyzovať a získavať informácie o jej vlastnostiach. Na základe týchto informácií je možné zvoliť najvhodnejší výrobný postup z hľadiska kvality výrobku a aplikovať ho do produkcie.

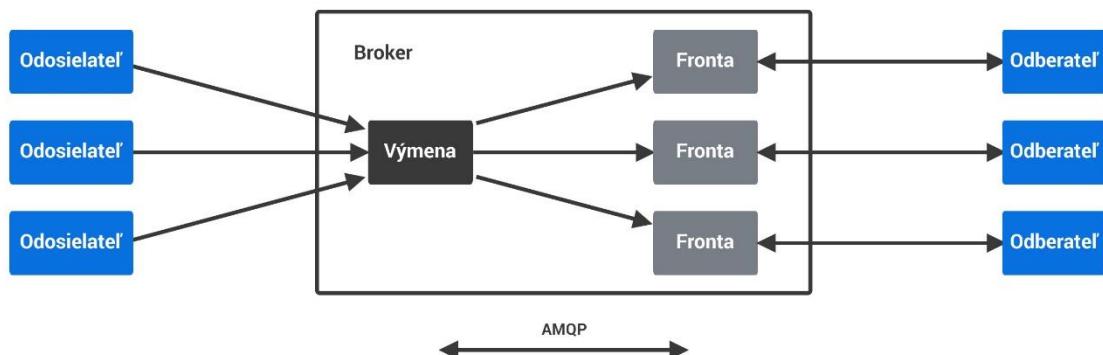
1.3.2 Komunikačné prostriedky IoT

Akonáhle sa dostávame do roviny komunikácie aplikácie a IoT zariadení, posúvajú sa nám hranice abstrakcie z načisto abstraktného sveta do sveta reálneho, a preto taktiež k tejto komunikácii musíme pristupovať odlišne ako ku komunikácii medzi dvomi aplikáciami. Na úrovni aplikačnej vrstvy rozlišujeme komunikačné protokoly na dva typy:

- **Žiadost – odpoveď (request - response)** je typický model pre počítače komunikujúce v sieti. V tomto modeli klientský počítač alebo softvér požaduje údaje alebo služby a serverový počítač alebo softvér reaguje na žiadost poskytnutím údajov alebo služby. Zjednodušene povedané, klient odošle požiadavku a server na ňu odpovedá správou späť. Medzi takéto komunikačné protokoly patrí napríklad HTTP, WebRTC alebo CoAP.

- **Publikovať – odoberať (publish - subscribe)**, alebo aj skrátene pub-sub, kde model komunikácie je nasledovný: centrálna jednotka, označovaná ako broker (niekedy označovaná aj ako server) prijíma a taktiež distribuuje všetky dátu. Pub-sub klienti v tomto prípade dát odosielajú, odoberajú, alebo oboje zároveň. Príkladom použitia tejto metódy sú protokoly ako XMPP, AMQP, DDS alebo MQTT pričom pre MQTT existuje taktiež aj zabezpečená verzia MQTT-SN.

AMQP je protokol aplikačnej vrstvy, ktorý používa medzi-prostriedok orientovaný na správu s významnými funkciami – smerovanie, bezpečnosť, orientácia správ, spoľahlivosť a radenie do fronty. Podobne ako protokol MQTT aj tento protokol používa mechanizmus publikovať – odoberať na prenos dát, ktorý poskytuje spoľahlivú komunikáciu a zaručuje doručovanie správ. Okrem toho AMQP podporuje rôzne funkcie heterogenity, interoperability, spoľahlivosti a bezpečnosti. Konkrétnie AMQP tiež udržuje brokera, aby poskytoval lepšiu spoľahlivosť prostredníctvom frontov, znižuje režijné náklady na strane klienta pomocou TCP pripojenia a poskytuje šifrovanie protokolmi TLS / SASL.



Obr. 8 AMQP komunikácia

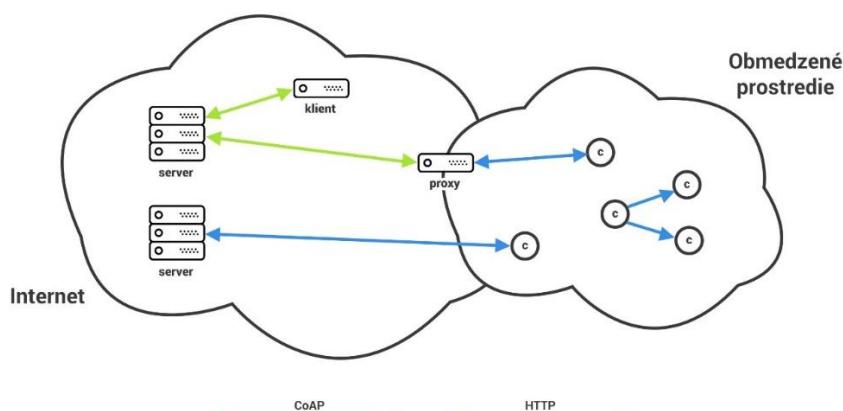
Obr. 8 znázorňuje architektúru AMQP, ktorá obsahuje tri komponenty: odosielateľ, broker a odberateľ. Komunikácia medzi odosielateľom a odberateľom bude výmenou za pomoci brokera. Odosielateľ je efektívny pri tvorbe a sprostredkovaní správ pre brokera. Broker poskytuje dve služby – jedna je výmena a druhá je udržiavanie fronty. Výmeny sa používajú na presmerovanie správ do špecifických frontov podľa vopred definovaných pravidiel. Zatiaľ čo fronty môžu ukladať správy a posielat ich odberateľom [9].

CoAP je odľahčený protokol, ktorý poskytuje komunikačný kanál a používa protokol UDP s typom správy žiadost – odpoveď. Je to jeden zo štandardných protokolov pre interakciu medzi rôznymi fyzickými zariadeniami. Pre dosiahnutie prenosu dát, CoAP udržuje veľkosť správy čo najmenšiu a podporuje mechanizmus zastavenia-čakania preposielania. Pre prenos správ pomocou CoAP, klienti sú priamo pripojení k serveru alebo klient bude pripojený k proxy, ktorý je pripojený na servery cez HTTP.

CoAP má štyri typy správ:

- s potvrdením
- bez potvrdenia
- rozpoznané
- obnoviteľné

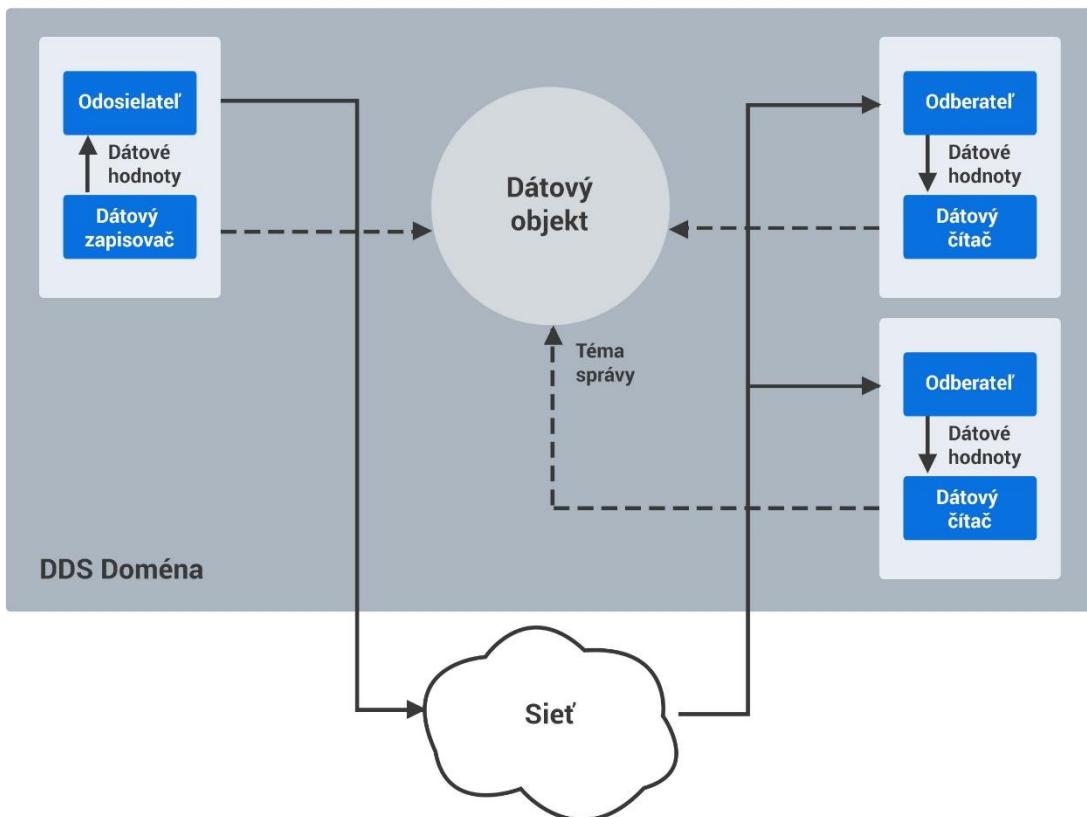
Navýše podporuje požiadavky M2M, DTLS, výmenu asynchronných správ a zisťovanie zdrojov, komunikáciu unicast a multicast. Rovnako ako HTTP, CoAP má model klient / server s dvoma vrstvami. Spodná vrstva je vrstva žiadost – odpoveď a horná vrstva je aplikačná vrstva. Aplikačná vrstva sa zaobrá rôznymi klientmi a externými aplikáciami. Kým vrstva žiadost – odpoveď pozostáva z rôznych prepojených serverov [9].



Obr. 9 CoAP komunikácia

Obr. 9 znázorňuje architektúru CoAP. Ako je uvedené, rozširuje normálnych HTTP klientov na klientov, ktorí majú obmedzenia zdrojov. Títo klienti sú známi ako klienti služby CoAP. Zariadenie proxy preklenuje medzeru medzi prostredím, v ktorom sa nachádzame, a typickým internetovým prostredím založeným na protokoloch HTTP. Ten istý server sa stará o protokolové protokoly HTTP aj CoAP [10].

Dátový protokol **DDS** bol navrhnutý skupinou OMG, ktorá sa používa hlavne pre komunikáciu M2M v rámci internetu, ktorá prebieha cez protokol UDP. Na rozdiel od iných protokolov používa obmedzený mechanizmus žiadost – odpoveď. Poskytuje však súbežnú a škálovateľnú výmenu údajov medzi odosielateľom a odberateľom. Okrem toho môže prenášať tisíce správ za jednotku času do niekoľkých prijímačov veľmi efektívne bez straty informácií. DDS poskytuje výnimočnú kvalitu služieb a spoľahlivosť tým, že podporuje 23 úrovňí QoS.



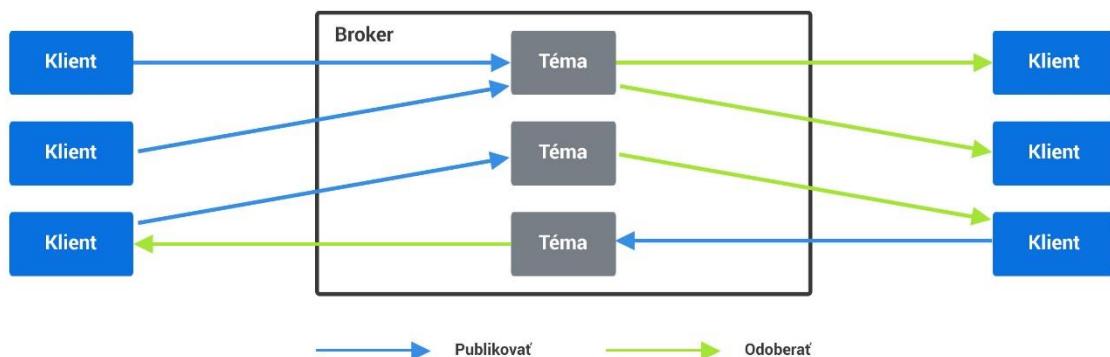
Obr. 10 DDS komunikácia

Na rozdiel od iných protokolov služba DDS má architektúru bez brokera. Obsahuje systém dátových objektov medzi odosielateľmi a odberateľmi. Títo odosielatelia a odberatelia sú prepojení cez sieť a téma správy súvisia so zapisovateľmi údajov a čítačmi údajov. Keď odosielateľ vysiela dátá, tak potom tam, kde sú údaje odovzdané na dátovom zapisovači, Obr. 10. Zatiaľ čo čítačky údajov sú schopné čítať a uchovávať údaje od rôznych používateľov [9].

MQTT podporuje komunikáciu M2M na základe komunikácie publikovať – odosielat', ktorá sa používa na pripojenie rôznych lokalít s malým priestorom a minimálnou šírkou pásma siete, Obr. 11. MQTT poskytuje flexibilitu pri pridávaní bezpečnostnej funkcie pre aplikácie na šifrovanie údajov, ktoré odosielajú a prijímanie, aby bol protokol ľahký a jednoduchý. MQTT pridáva významné zaťaženie siete pri riešení problému zabezpečenia pomocou vrstvy soketu (SSL), ktorá už odľahčená nie je. MQTT sa zameriava predovšetkým na tri vlastnosti QoS:

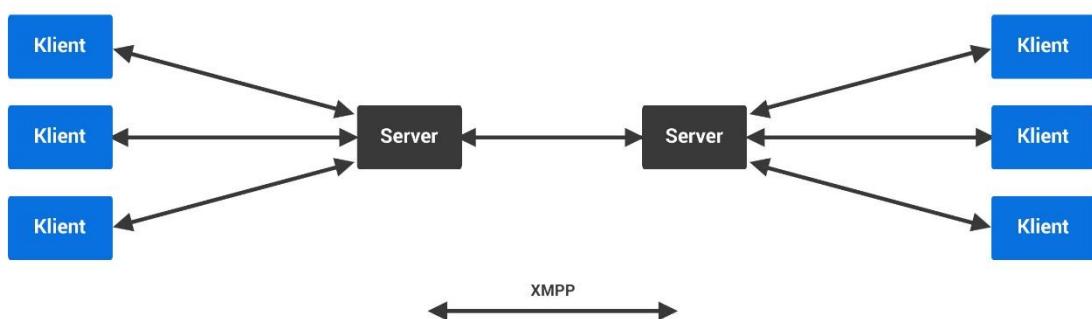
- QoS 0 - "nanajvýš raz doručené". Používateľ môže odoslať správu najrýchlejším spôsobom bez čakania na odpovede.
- QoS 1 - "aspoň raz doručené". Klient alebo server by mali poslat' aspoň jednu správu bez ohľadu na duplicitné správy.
- QoS 2 - "presne jedenkrát doručené". Správy sa prenášajú iba raz bez toho, aby sa umožnili duplicitné správy.

Okrem toho protokol MQTT zabezpečuje najlepšiu komunikáciu v mobilných aplikáciach kvôli svojej malej veľkosti, efektívnej distribúcii informácií jednému alebo viacerým prijímačom, minimalizácii dátových paketov a nízkej spotrebe energie [9].



XMPP je jedným z populárnych protokolov známych ako "Jabber", čo je open source protokol pre okamžité odosielanie správ, ktorý spája rôznych ľudí pomocou textových správ. Konkrétnie sa tento protokol skladá zo služby XML, ktorá využíva protokol TCP na spoľahlivú komunikáciu. Navyše, XMPP má zabudované zabezpečenie a prispôsobuje súčasné a budúce aplikácie, ktoré chýbajú v iných základných protokoloch. Zabudovaná bezpečnostná funkcia XMPP využíva mechanizmus SASL aj TLS na zabezpečenie integrity údajov, overovania a komunikácie. Zabezpečenie a overenie totožnosti sa dá dosiahnuť technikou SASL zatiaľ čo na bezpečnú komunikáciu medzi aplikáciou a aplikáciou sa používa mechanizmus zabezpečenia TLS.

Komponenty dostupné v architektúre XMPP sú predovšetkým servery a klienti. Smerovanie poskytuje server na prenos dát z jedného jedinečného klienta na iného jedinečného klienta a tiež pre cudzie domény prostredníctvom brán [9].



Obr. 12 XMPP komunikácia

Tabuľka 2 Porovnanie základných vlastností IoT komunikačných protokolov [9]

<i>Protokol</i>	<i>QoS</i>	<i>Zabezpečenie</i>	<i>Šírka pásma</i>	<i>Caching</i>	<i>Objavenie zdrojov</i>	<i>Typ služby</i>
<i>AMQP</i>	Áno	SSL	Vysoká	Áno	Nie	TCP
<i>CoAP</i>	Áno	DTLS	Nízka	Áno	Áno	UDP
<i>DDS</i>	Áno	SSL, DTLS	Nízka	Áno	Áno	UDP
<i>MQTT</i>	Áno	SSL	Nízka	Áno	Nie	TCP
<i>XMPP</i>	Nie	SSL	Nízka	Áno	Áno	TCP

Tabuľka 3 Sumár IoT protokolov [9]

<i>Protokol</i>	<i>Charakteristika</i>	<i>Postup</i>	<i>Výhody</i>	<i>Nevýhody</i>	<i>Nasadenie</i>
<i>AMQP</i>	Radí správy do fronty a je interoperabilný.	Navrhnuté na podporu širokej škály správ a komunikačných vzorov	Vysoko spoľahlivý, dopredná a ukladaná komunikácia	Pracuje len pri vyšších šírkach pásma	Biznis orientované odosielanie správ, bankový priemysel
<i>CoAP</i>	Synchrónna požiadavka na odpoved', 1-1 alebo M-M	Používa sa v jednoduchých elektronických zariadeniach, ktoré im umožňujú komunikovať interaktívne cez siet'.	1-1 Komunikácia, M-M komunikácia, Zisťovanie zdrojov	Menej štandardné, zrelšie a štandardné v porovnaní s MQTT.	Inteligentné domy, inteligentná sieť a Automatizácia budov
<i>DDS</i>	Interoperabilná dátová služba s vysokým výkonom	Na pripojenie zariadenia k inému zariadeniu a tiež zdieľanie správnych údajov na správnom mieste	interoperabilný, šetrí šírku pásma, flexibilný a spoľahlivý	Nemajú žiadnu škálovateľnosť	Vojenské systémy, Integrácia nemocní a veterné farmy.
<i>MQTT</i>	Nízka spotreba energie, M-M komunikácia	Pub-Sub hlavným cieľom je zhromažďovanie údajov a prenos do IT infraštruktúry	Úspora energie a pamäte, nízka spotreba energie	Dlhodobé pripojenie TCP, názvy tém sú dlhé reťazce	Domáca automatizácia, aplikácie na úrovni podnikov
<i>XMPP</i>	Šifrovanie kanálov a kontrola prítomnosti	Umožňuje používateľom internetu odosielat' okamžité správy	Bezpečný, vyhľadávanie služieb, veľmi robustný, výkonný	Tok údajov je viac ako server XMPP, chýba celosvetová podpora	Okamžité odosielanie správ, Skupinový rozhovor, Hranie, Sledovanie vozidiel

1.3.3 Typy implementácie IoT

V konečnom dôsledku by sa dali typy implementácie IoT technológie rozdeliť do troch hlavných skupín [11]:

- **Spotrebiteľský IoT** – Zariadenia a služby spotrebiteľského internetu sú zamerané na individuálnych užívateľov alebo rodiny. Množstvo zariadení bude jedinečné pre každého spotrebiteľa, hoci niektoré môžu byť pre viac užívateľov naraz ako napr. niekoľko termostatov pre väčší dom alebo inteligentné osvetlenie, kde každá žiarovka je zariadenie. Tento obmedzený počet zariadení má tendenciu zjednodušiť softvér používateľského rozhrania pre správu z rôznych dôvodov, vrátane bezpečnosti. Produkty sú často izolované na niekoľko používateľov, ktoré môžu mať obmedzené, ak nejaké, prístupové ovládacie prvky okrem prístupu k zariadeniam pomocou zodpovedajúcich aplikácií. Z pohľadu používateľa je kolekcia zariadení a ich správa izolovaná pre daného používateľa alebo jeho rodinu.
- **Komerčný IoT** – sa nachádza medzi spotrebiteľmi a priemyselným internetom a zdieľa aspekty oboch. Niektorí môžu vlastne zoskupiť obchodné a priemyselné internetové spojenie. Rozsah komerčného internetu vecí sa obvykle nachádza na úrovni spoločnosti alebo organizácie. Aplikácie ako inteligentná energia a osvetlenie by spadali pod komerčný IoT. Jedným z príkladov komerčnej platformy IoT je systém Verdigris Einstein. Zapája sa do existujúcej infraštruktúry budov, aby poskytovala inteligentné meranie a monitorovanie s cieľom minimalizovať spotrebu elektrickej energie vo veľkých komerčných budovách, ktoré môžu mať rôznorodých zamestnancov. Systém Verdigris poukazuje na niektoré veľké rozdiely medzi spotrebiteľskými a komerčnými systémami internetu. Prvý je rozsah integrácie, po ktorej nasleduje správa a analýza. Komerčné systémy zriedka zahŕňajú jednotlivé zariadenia - v závislosti od aplikácie môžu byť desiatky alebo tisíce. V jednom systéme sa môže zapojiť aj niekoľko lokalít, ktoré sa pridajú k výzve manažmentu.
- **Priemyselný IoT** – môže byť považovaný za robustný, dlhodobý komerčný Internet vecí, ktorý však prehliada rozdiely v dizajne a infraštrukture IIoT. Rovnako ako mnohé komerčné riešenia, riešenia IIoT sa často zameriavajú na existujúce automatizované priemyselné systémy. Rozdiel je v tom, že tieto systémy môžu byť staršie, takže úroveň senzorov je často založená na tom, čo je k dispozícii. Poskytujú dostatočné informácie na kontrolu priemyselného procesu, ale ďalšie informácie by boli užitočné, ak by bolo

možné začleniť viac senzorov do procesu. Takéto snímače môžu sledovať stav komponentov. Niekedy môže poskytnúť dodatočné informácie o opotrebovaní systému a predvídať požiadavky na údržbu.

1.4 Digitálne dvojča

Digitálne dvojča (DT – Digital Twin) v pôvodnej podobe je opísané ako digitálny model informácií o fyzickom systéme vytvorenom ako entita ako taká a spojená s daným fyzickým systémom. Digitálne zastúpenie by malo optimálne zahŕňať všetky informácie týkajúce sa systému, ktoré by sa mohli potenciálne získať z dôkladnej kontroly v reálnom svete.

Hlavným aspektom DT je schopnosť poskytovať rôzne informácie v konzistentnom formáte a reálnom čase. Digitálne dvojčatá sú viac ako len čisté údaje, obsahujú algoritmy, ktoré opisujú ich skutočný náprotivok a rozhodujú o krodoch vo výrobnom systéme na základe týchto spracovaných údajov [12].



Obr. 13 Vývoj Digitálneho dvojča

Obr. 13 zobrazuje štyri fázy vývoja Digitálneho dvojča:

- **Prvá fáza** – celý výrobný proces existuje len ako fyzická verzia.
- **Druhá fáza** – pridáva sa digitálna verzia, ktorá doplní fyzickú verziu o ďalšie informácie.
- **Tretia etapa** – začína interakcia medzi fyzickou a digitálnou verziou.
- **Štvrtá etapa** – existujú ďalšie interakcie a konvergencia medzi fyzickou a digitálnou verziou.

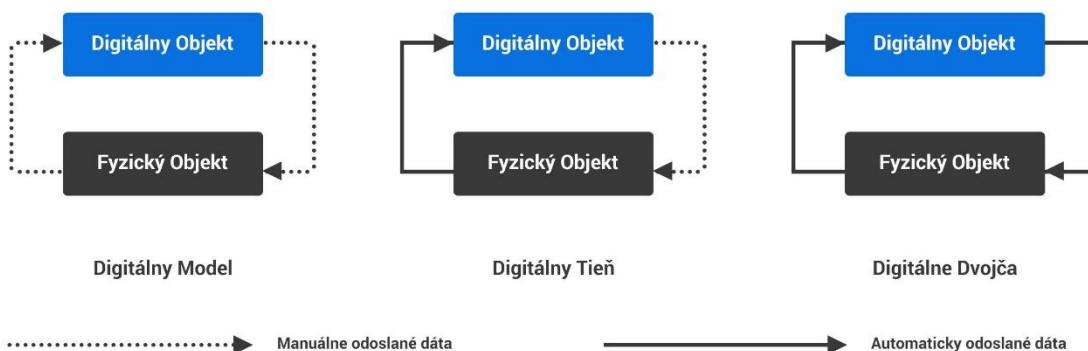
Čo ale robí Digitálnym dvojča? Čím je? Digitálne dvojčatá využívajú historické údaje o kontexte a činnosti na pochopenie minulosti, používajú priame a nepriame údaje na

zobrazenie súčasných podmienok a používajú strojové učenie a znalosti na predpovedanie budúcnosti stavu a správania sa príslušného fyzického objektu. Spoločnosť GE Digital definuje hierarchiu digitálnych dvojčiat ako celok štyroch hlavných zložiek:

- **Komponent** – alebo taktiež súčiastka, ako je napríklad ložisko na rotujúcom zariadení. Táto zložka je zvyčajne významnou súčasťou objektu, ktorá má významný vplyv na efektívnosť objektu, ku ktorému patrí.
- **Objekt** – je celkový objekt, ktorý digitálne dvojča projektuje. Napríklad motor alebo čerpadlo. Digitálne dvojčatá k takýmto objektom môžu byť celkom zostaveným z viacerých komponentov, pričom tento zrkadlený objekt poskytuje projekciu na úrovni zariadenia.
- **Systém** – alebo tiež jednotka, je súbor objektov, ktoré spoločne tvoria systémovú alebo sietiovú funkciu, ako napríklad ropnú a plynovú rafinériu alebo výrobnú linku. Takéto dvojča poskytuje viditeľnosť a pohľad do množstva vzájomne závislých zariadení.
- **Proces** – je zvyčajne dvojča najvyššej úrovne, ktoré poskytuje obraz súboru činností alebo operácií, ako je výrobný proces. Procesné dvojča môže byť informované súborom dvojčiat objektov alebo systémov, ale zameriava sa viac na samotný proces a nie na zariadenie.

Vďaka poznaniu aktuálneho stavu a predpovedaniu budúceho stavu DT je možné účinne monitorovať, simulovalať a riadiť aktíva alebo proces a optimalizovať životné cykly [13].

Na základe uvedených definícii Digitálneho dvojča v akomkoľvek kontexte by bolo možné charakterizovať spoločné chápanie Digitálnych dvojčiat ako digitálnych náprotivkov fyzických objektov. V rámci týchto definícii sú výrazy Digitálny Model, Digitálny Tieň a Digitálne Dvojča často synonymom. Avšak uvedené definície sa líšia úrovňou integrácie dát medzi fyzickým a digitálnym partnerom. Niektoré digitálne reprezentácie sú manuálne modelované a nie sú spojené so žiadnym existujúcim fyzickým objektom, zatiaľ čo iné sú plne integrované do výmeny údajov v reálnom čase. Na základe tejto skutočnosti je podľa štúdie [12] možné rozdeliť Digitálne dvojičky podľa úrovne integrácie dát nasledovne:



Obr. 14 Tri rozdelenia Digitálnych Dvojčiat podľa úrovne integrácie

Digitálny Model je digitálne znázornenie existujúceho alebo plánovaného fyzického objektu, ktorý nepoužíva žiadnu formu automatickej výmeny údajov medzi fyzickým objektom a digitálnym objektom. Digitálne zobrazenie môže obsahovať viac-menej komplexný opis fyzického objektu. Tieto modely môžu zahrňať, ale nie sú obmedzené na simulačné modely plánovaných tovární, matematické modely nových produktov alebo iné modely fyzického objektu, ktoré nepoužívajú žiadnu formu automatickej integrácie údajov. Digitálne dátá existujúcich fyzických systémov sa môžu stále používať na vývoj takýchto modelov, ale všetka výmena údajov sa vykonáva ručne. Zmena stavu fyzického objektu nemá priamy vplyv na digitálny objekt a naopak.

Digitálny Tieň je definovaný rovnako ako Digitálny model s výnimkou jednosmerného automatického toku dát medzi fyzickým objektom a objektom digitálnym, pričom automatický tok informácií smeruje z fyzického objektu do objektu digitálneho. To znamená, že každá zmena fyzického objektu je automaticky projektovaná do jeho digitálneho modelu.

Digitálne Dvojča je digitálny objekt, ktorý automaticky zobrazuje nielen zmeny fyzického modelu, ale taktiež sú automaticky posielané zmeny digitálneho modelu na fyzický, čím sa digitálny objekt stáva nielen zrkadlom fyzického objektu, ale taktiež jeho riadiacou jednotkou.

S rastúcim nasadením internetu vecí koncept digitálnej verzie každej fyzickej veci získal svoju dôležitosť. Preto je možné **digitálne dvojča aplikovať** napríklad **do oblastí** ako je komerčná oblasť, ladenie výkonu, digitálna výroba zariadení, zdravotníctvo, inteligentné mestá alebo údržba.

V prípade **konzumentskej oblasti** zákazníci zohrávajú kľúčovú úlohu pri ovplyvňovaní stratégií a rozhodnutí v každej firme. Konečným cieľom každej organizácie je získať a udržať si veľkú zákaznícku základňu, čo znamená zlepšenie skúseností zákazníka. Digitálne dvojča môže pomôcť zvýšiť služby priamo ponúkané zákazníkom. Napríklad digitálne dvojičky by mohli byť použité na modelovanie módy na vizuálnej kópii zákazníka.

Pri **ladení výkonu** Digitálne dvojča pomáha určiť optimálny súbor akcií, ktoré môžu pomôcť maximalizovať niektoré kľúčové ukazovatele výkonnosti a tiež poskytnúť prognózy pre dlhodobé plánovanie. Napríklad, výkon vedeckého zariadenia, ktorý je nasadený na kozmickej lodi, možno vyladiť zo Zeme pomocou digitálnej 3D vizualizácie v reálnom čase.

Digitálne dvojča môže byť použité ako digitálna kópia skutočného stroja, vytvorená a vyvíjaná súčasne s **výrobou zariadení**. Vezmime si príklad nemeckého výrobcu strojov, ktorý sa rozhodol digitálne mapovať obaly a špeciálne stroje, ktoré vyrába pre mnohé odvetvia. Údaje o skutočnom stroji boli vložené do digitálneho modelu predtým, než začala skutočná výroba. Takže dvojitý digitálny systém umožňuje simuláciu a testovanie nápadov ešte pred začiatkom samotnej výroby.

Ďalšie využitie môže byť aplikované do odvetvia **zdravotníctva**, kde toto dvojča pomáha virtualizovať nemocničný systém s cieľom vytvoriť bezpečné prostredie a otestovať vplyv možných zmien na výkon systému. Nielen operácie, digitálne dvojčatá môžu tiež pomôcť zlepšiť kvalitu zdravotníckych služieb poskytovaných pacientom. Napríklad chirurg môže použiť digitálnu dvojičku na digitálnu vizualizáciu srdca, predtým ako ho operuje.

Inteligentné mestá. DT môže byť použité na zachytenie priestorových a časových dôsledkov na optimalizáciu udržateľnosti miest. Napríklad "Virtual Singapur", ktorá je súčasťou iniciatívy singapurskej vlády Smart Nation Singapore, je prvým DT existujúceho mestského štátu na svete, ktorý poskytuje Singapurčanom efektívny spôsob zapojenia sa do digitálnej ekonomiky.

DT je schopné analyzovať údaje o výkonnosti získané v priebehu času a za rôznych podmienok. Napríklad digitálnou kópiou motora môže byť zobrazený motor pretekárskych vozidiel s cieľom identifikovať požadovanú **údržbu**, ako napríklad komponent, ktorý má tendenciu časom zhriet.

2 Umelá inteligencia a umelé neurónové siete

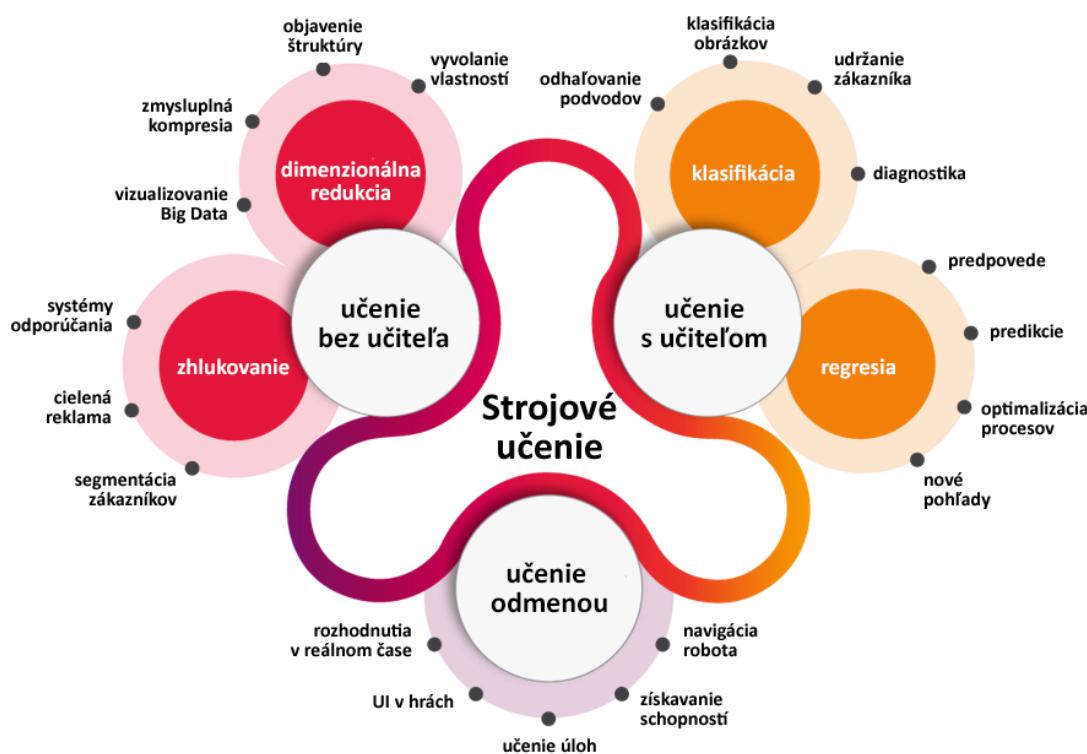
V druhej kapitole dizertačnej práce sa zameriavame na základné teoretické východiská inteligentných algoritmov a definíciu základných pojmov. Venujeme sa princípom umelých neurónových sietí, ktoré ďalej rozvádzame na hlboké učenie, teóriu konvolučných neurónových sietí a rekurentných neurónových sietí. Opisujeme základné postupy a prístupy ku konfiguráciám umelých neurónových sietí.

Umelá inteligencia je pojem, ktorý sa čoraz viac objavuje v povedomí ľudí, a ktorý sa teší rastúcemu trendu. Tento termín mal donedávna nejasný význam, kedy sa názory na to, čo vlastne umelá inteligencia je, rozchádzali. Samotné označenie umelá inteligencia však prešla formalizáciou a je to „simulácia ľudského správania strojom, v prípade zefektívnenia stroja na identifikáciu a použitie správnej časti vedomosti v danom kroku riešenia problému“. Stroj, ktorý má schopnosť plánovania a vykonávania správnej úlohy v správnom čase je vo všeobecnosti nazývaný ako racionálny stroj. Preto je umelá inteligencia taktiež označovaná aj ako „oblasť“, ktorá sa zaoberá výpočtovými modelmi schopnými myslieť a konáť racionálne“. Na to, aby systém vedel reagovať racionálne, potrebuje získať vedomosti z reálneho sveta; potrebuje vnímať, potrebuje získavať informácie, potrebuje sa učiť [14].

Idea a terminológia existuje už dlhšie. V roku 1956 sa termín umelá inteligencia prvýkrát objavil na Dartmouth College pod vedením profesora J. McCarthy-ho. Odvtedy umelá inteligencia prešla niekoľkými pádmi a vzostupmi vzhľadom na vyvíjajúce sa technológie [15]. Samotná umelá inteligencia je označovaná aj ako „deep learning“ alebo hlboké učenie. Toto označenie je známe niečo viac ako poslednú dekádu, predtým pomenúvané aj ako kybernetika (1940-1960) alebo konektivizmus (1980-1990). Jedny z prvých učiacich sa algoritmov predstavovali výpočtový model založený na báze toho, ako k učeniu dochádza alebo by mohlo dochádzať v samotnom mozgu. Na tomto princípe vznikli umelé neurónové siete. Umelé neurónové siete boli niekedy využívané na pochopenie funkcie mozgu, no predovšetkým na tvorbu algoritmov strojového učenia [16].

Aj napriek tomu, že základy strojového učenia postaveného na umelých neurónových sietach boli položené, úroveň technických prostriedkov danej doby neumožňovala nasadenie ich potenciálu do praxe. Spočiatku bol problém so slabou výpočtovou silou, malou kapacitou databáz či rýchlosťou pripojenia. Výskum sa začal orientovať na riešenie konkrétnych problémov ako napríklad rozpoznávanie obrazu či reči, no postupom času chýbala

škálovateľnosť'. Nástupom hlbokého učenia ako ho poznáme dnes, výrazným nárastom objemu dostupných informácií, zvýšenia výpočtovej sily či konektivity sa umelá inteligencia pohla vpred. V dnešnej dobe je možné aplikovať princípy strojového učenia na väčšinu oblastí života; či už ide o finančníctvo, medicínu, reklamu, zábavu, poľnohospodárstvo či techniku. V priemyselnej oblasti má umelá inteligencia veľký potenciál a priestor na rozvoj. V kontexte Industry 4.0 sa jedná predovšetkým o inteligentnú výrobu, kedy jednotlivé stroje výrobnej linky dokážu spracovať osobitné procesy na základe naučených princípov. Je možné naučiť stroj, aby rozoznával chybné diely a vyradil ich z výroby. Ďalším príkladom je automatické vyhodnocovanie stavu stroja, predikcia potrebnej údržby alebo rozpoznávanie a triedenie jednotlivých komponentov na výrobnej linke.



Obr. 15 Strojové učenie podľa zaradenia

Ked' hovoríme o strojovom učení, rozoznávame 3 základné typy [17]:

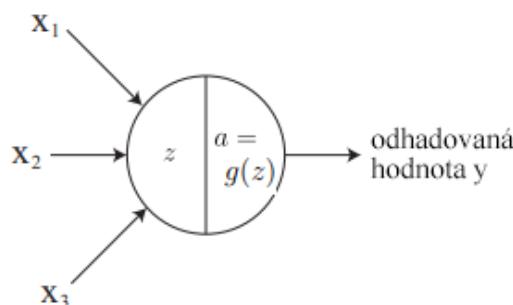
- **učenie s učiteľom** – stroj sa učí na základe vopred označených dát. Využíva sa na klasifikáciu a regresiu ako je napríklad klasifikácia obrázkov, diagnostika, predpoved, predikcia alebo optimalizácia procesu.

- **učenie bez učiteľa** – stroj sa učí spoločné charakteristiky a na ich základe objekty združuje do rovnakých skupín. Používa sa na zgrupovanie objektov alebo redukciu ako je napríklad cielená kompresia, odhalovanie štruktúry, vizualizácia Big data či segmentácia zákazníkov alebo cielená reklama.
- **učenie odmenou** – stroj sa učí na základe odozvy. Príkladom je rozhodovanie v reálnom čase, navigácia alebo získavanie schopností.

2.1 Princíp umelých neurónových sietí

Základným prístupom strojového učenia, je použitie neurónovej siete (NN). Umelé NN (ANN) je vzájomne prepojená skupina uzlov inšpirovaná zjednodušením neurónov v mozgu. Existuje niekoľko architektúr a techník na aplikovanie ANN, ako sú konvolučné neurónové siete (CNN) na získanie od najmenších až po najväčšie vlastnosti, rekurentné neurónové siete (RNN) na opakujúce sa udalosti a najmä na hlboké učenie v kontexte neurónových sietí (DNN) odkazujúce na neurónovú sieť s viacerými skrytými vrstvami. Všetky tieto techniky a mnohé ďalšie zabezpečujú vytvorenie komplexných neurónových sietí na prekonanie mnohých zložitých problémov.

Podstatou neurónovej siete v zjednodušenom tvaru je transformácia vstupných údajov na výstupné údaje, $f : x \rightarrow y$. Na ilustráciu uvažujme jedno-neurónovú sieť, kde je neurón definovaný **vstupnou vrstvou** s 3 vstupnými údajmi x_1, x_2, x_3 , **skrytou vrstvou** s parametrizovaným vstupom z a výsledkom tzv. aktivačnej funkcie a a **výstupnou vrstvou** s 1 výstupným údajom y :



Obr. 16 jednoduchá neurónová sieť

kde parametrizovaný vstup z predstavuje zmenu vstupného vektora x vzhľadom na vektor váh spojenia W a skalárny bias b nasledovne:

$$z_j^{[l]} = W_j^{[l]T} x + b_j^{[l]} \quad (1)$$

pričom $z, x, w, b \in R$ a index l označuje l -tu vrstvu a index j označuje j -tu jednotku vo vrstve. Výsledok aktivačnej funkcie a závisí od parametrizovaného vstupu z nasledovne:

$$a_j^{[l]} = g(z_j^{[l]}) \quad (2)$$

kde funkcia g je aktivačná funkcia nelineárneho charakteru. Samotný výstup y je reprezentovaný ako výsledok aktivačnej funkcie jednotiek predošej vrstvy a teda vektora $a^{[l-1]}$:

$$a^{[l]} = \begin{bmatrix} a_1^{[l-1]} \\ \vdots \\ a_j^{[l-1]} \end{bmatrix} \quad (3)$$

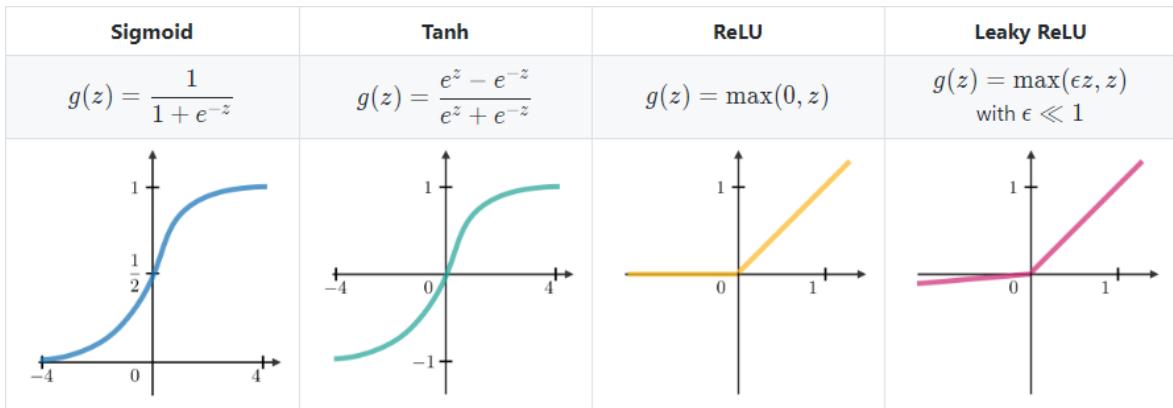
Vzhľadom od charakteru očakávaného výstupu je zvolená aj aktivačná funkcia na výstupnej vrstve [18].

Vnútri v sieti sa na trénovanie využíva predovšetkým **ReLU** aktivačná funkcia, prvýkrát predstavená v roku 2000, no svoj úspech zožala predovšetkým pri aplikácií do hlbokých neurónových sietí v roku 2012. ReLU aktivačná funkcia bola jedným z hnacích prvkov, ktoré výrazne zvýšili záujem o umelú inteligenciu. Vďaka nej sa hlboké neurónové siete optimalizujú jednoduchšie ako za použitia iných aktivačných funkcií [19].

Výsledok ReLU aktivačnej funkcie sa pohybuje iba v nezáporných hodnotách. Pokial' je parametrizovaný vstup záporný, výsledkom operácie je 0, inak, je výsledkom samotný vstup. Zapisujeme ako:

$$g(z) = \max(0, z) \quad (4)$$

V praxi sa popri ReLU častokrát využíva aj sigmoid, tanh alebo LReLU.



Obr. 17 Porovnanie aktivačných funkcií sigmoid, tanh, ReLu, LReLU [20]

Po vzniku a nasadení ReLU aktivačnej funkcie do praxe vzniklo aj niekoľko aktivačných funkcií, ktoré boli ReLU inšpirované a vykazujú lepšie výsledky [19].

V rámci procesu trénovania siete postupne tieto hodnoty vkladáme na vstup a získavame predpovede na výstup. Model neurónovej siete pozostáva z dvoch hlavných časťí:

- **architektúra siete**, definovaná počtom vrstiev, počtom jednotiek vo vrstvách a spôsobom prepojenia vrstiev
- **parametre siete**, hodnoty známe taktiež ako váhy w a bias b

Trénovanie siete je vlastne numerická optimalizácia parametrov siete. Než začneme siet' trénovať, je potrebné tieto parametre inicializovať. Spravidla sa inicializujú v hladinách $(0, 0.1)$, aby neboli nulové a pri ich aktualizácii sme sa vedeli pohnúť ďalej, no zároveň aby mali dostatočne malú hodnotu a gradienty nenarastli do príliš vysokých hodnôt, tzv. „nevybuchli“. Gradient v prípade neurónových sietí opisuje zmenu parametrov siete v každom bode siete a určuje akým smerom sa siet' trénuje. V rámci procesu trénovania umelých neurónových sietí predpokladáme, že nemáme iba jeden vektor vstupných hodnôt, to znamená, že nemáme iba jeden konkrétny prípad nášho problému, ale že máme takýchto vzoriek viac. Na to, aby sme vedeli gradient vypočítať a aktualizovať váhy siete, potrebujeme po predikovaní jednej vzorky získať hodnotu chybovej funkcie, na základe ktorej aktualizujeme váhy siete a tým samotnú siet' trénujeme. Chybová funkcia nám určuje, nakoľko sa predikovaná hodnota na výstupe líši od skutočnej a takúto funkciu označujeme ako:

$$L(\theta) = L(\hat{y}, y) \quad (5)$$

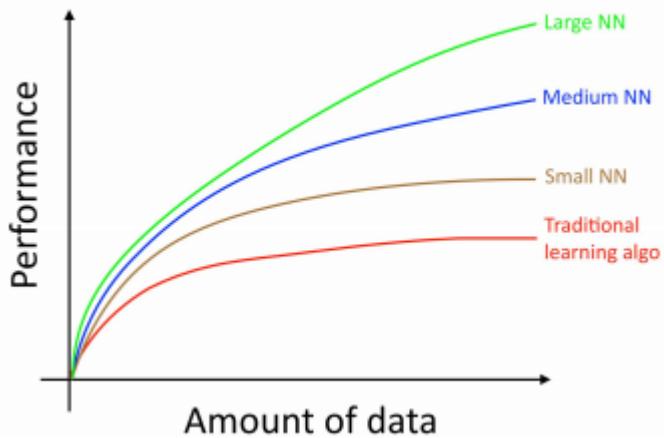
kde L je naša chybová funkcia, θ sú parametre, \hat{y} je predpovedaná hodnota a y je skutočná hodnota. Výsledkom takejto funkcie je skalárna hodnota, ktorú sa snažíme minimalizovať naprieč iteráciami alebo taktiež aj nazývanými epochami učiaceho procesu. Pomocou hodnoty chybovej funkcie vieme vypočítať gradient vzhľadom na parametre siete θ , W a b , ako $\partial L / \partial W$ a $\partial L / \partial b$ a tak ich aktualizovať ako odpočítanie násobku α hodnoty určujúcej rýchlosť učenia a derivácie chybovej funkcie podľa daného parametru $\partial L / \partial \theta$ od ich pôvodnej hodnoty θ v konkrétnej vrstve siete l :

$$\theta^{[l]} = \theta^{[l]} - \alpha \frac{\partial L}{\partial \theta^{[l]}} \quad (6)$$

Takému algoritmu učiaceho procesu taktiež hovoríme aj spätné šírenie chyby a bol jednou z ideí, ktorá vzbudila nádej v neurónové siete a pomohla posunúť ich efektivitu a robustnosť na vyšší level. Bez algoritmu spätného šírenia chyby museli byť parametre siete nastavované manuálne, čo v prípade hlbokých neurónových sietí s miliónmi parametrov neprichádza do úvahy [18] [21].

2.2 Hlboké neurónové siete

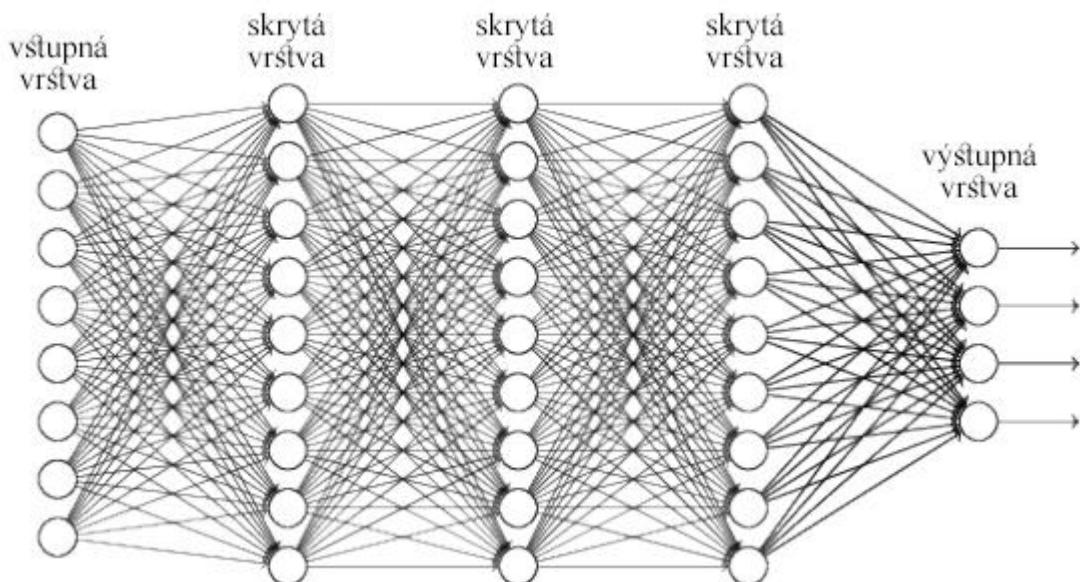
Podstatou hlbokých neurónových sietí je, že majú viac ako 1 skrytú vrstvu. Jednotlivé uzly siete sú napojené na uzly iných vrstiev siete. V prípade hlbokých neurónových sietí – DNN (Deep Neural Networks) sa siet dokáže naučiť jemnejšie znaky, potrebné na určenie čo najpresnejšieho predpokladu než siete iba s 1 skrytou vrstvou. DNN, na to aby využili svoj potenciál, potrebujú aj väčší objem dát, na ktorých sa trénujú. Objem dát vzhľadom na digitalizáciu mnohých procesov a informácií v dnešnej dobe nie je problém a tak sa dokáže siet učiť efektívnejšie ako kedykoľvek predtým v porovnaní s menšími architektúrami, či tradičnými algoritmami, ako nám udáva aj Obr. 18.



Obr. 18 Výkon NN vzhľadom k objemu dát a veľkosti jej architektúry [22]

Jednotlivé uzly siete je možné rozdeliť do 5 hlavných kategórií, podľa ich role v sieti [23]:

- **Vstupný uzol** – Uzol vstupnej vrstvy siete.
- **Dopredný uzol** – Uzol skrytej vrstvy siete je tradičným uzlom sieti s doprednou funkciou. Uzol je pripojený k ostatným neurónom váhami.
- **Výstupný uzol** – Uzol udávajúci výstup NN.
- **Rekurentný uzol** – Uzol s pamäťovou jednotkou, uchovávajúcou predoslový stav uzlu.
- **Konvolučný uzol** – Uzol s doprednou funkciou, uchovávajúci priestorové informácie.



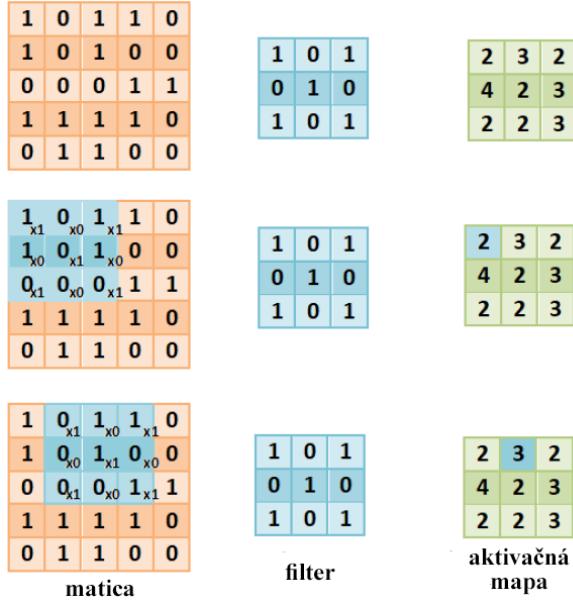
Obr. 19 Plne prepojená hlboká neurónová sieť

Obr. 19 znázorňuje siet' so vstupnou vrstvou, 3 skrytými vrstvami a výstupnou vrstvou, kde všetky uzly sú prepojené s uzlami predošej a nasledujúcej vrstvy. Takejto sieti hovoríme, že je **plne prepojená**. Plne prepojená siet' si vyžaduje značne viac výpočtových zdrojov ako čiastočne prepojená siet', nakoľko je potrebné počítať gradient v každom prepojení a čím hlbšia siet' je, tým je v nej viac prepojení, ktoré z celkového hľadiska často nemajú dostatočné opodstatnenie, no výrazne spomaľujú učiaci proces.

2.3 Konvolučné neurónové siete

Konvolučné siete, známe tiež ako konvolučné neurónové siete alebo CNN, sú špecializovaným druhom neurónovej siete na spracovanie údajov, ktorá má známu topológiu podobnú mriežke. Tieto siete sú využiteľné pri spracovávaní časovo orientovaných dát, ktoré možno považovať za 1D mriežku predstavujúcu vzorky v pravidelných časových intervaloch, a obrazové údaje, ktoré možno považovať za 2D mriežku pixelov. Konvolučné siete boli v praktických aplikáciách nesmierne úspešné. Názov „konvolučná neurónová siet“ označuje, že siet používa matematickú operáciu nazývanú konvolúcia. Konvolúcia je špecializovaný druh lineárnej operácie. Konvolučné siete sú jednoducho neurónové siete, ktoré používajú konvolúciu namiesto všeobecného násobenia matíc v aspoň jednej zo svojich vrstiev.

Úlohou konvolúcie v CNN je vybrať hlavné vlastnosti zo vstupnej matice. Konvolúcia zachováva vzťahy jednotlivých bodov v priestore. Princíp konvolúcie pozostáva z násobenia vstupnej matice s filtrom určitej veľkosti po jednotlivých prvkoch a ich následnom sčítaní, čoho produkтом je aktivačná mapa nazývaná takiež mapa vlastností. Pri spracovávaní matice sa filtre pohybujú po celej šírke a výške vstupnej matice. V prípade trojrozmerného objektu, tenzoru, akým je aj RGB obrázok, sa jednotlivé filtre aplikujú rovnako pre každý kanál obrázka – R, G aj B. Ukážka je na Obr. 20. V prvej tretine obrázka je vyobrazená vstupná matica (oranžová), filter (modrá) a produkt operácie konvolúcie – aktivačná mapa (zelená). V druhej časti obrázka je vyobrazená aplikácia filtra na vstupnú maticu v prvom kroku a výsledná bunka aktivačnej mapy prislúchajúca danej operácii. V tretej časti je vyobrazený druhý krok, kedy sa konvolučný filter na matici posunul o jeden stĺpec doprava a tak bola vypočítaná druhá bunka aktivačnej mapy.



Obr. 20 Konvolučná operácia

Na veľkosť mapy vlastností konvolúcie majú vplyv tri parametre: hĺbka, krok filtra a okraj. V závislosti od počtu filtrov použitých v konvolúcii sa hĺbka (počet kanálov) výstupu zvyšuje. Krok filtra hovorí, o koľko prvkov má byť filter posunutý ďalším násobením matíc. Niekedy je vstupná matica doplnená nulami k okraju, takže výsledná mapa vlastností môže mať rovnakú šírku a výšku ako vstupná matica. Veľkosť aktivačnej mapy môžeme vypočítať pomocou vzťahu:

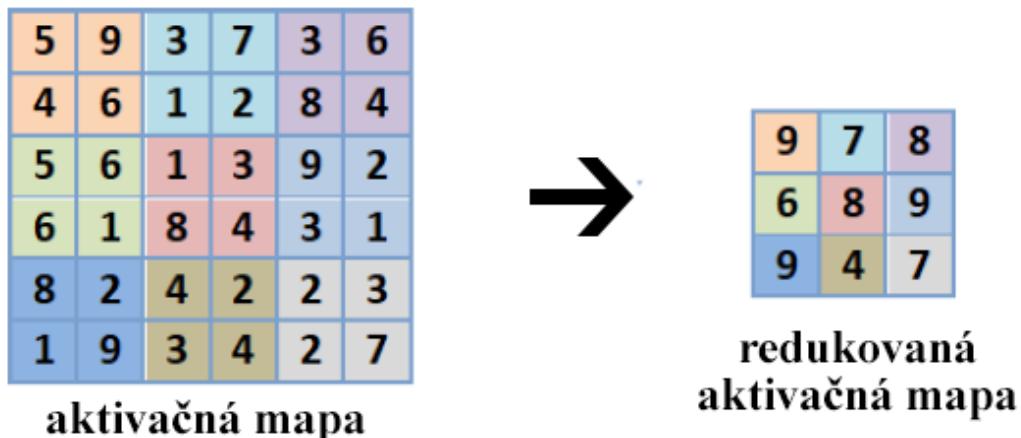
$$N = \frac{W - F + 2P}{S} + 1 \quad (7)$$

kde N označuje veľkosť aktivačnej mapy, W je veľkosť vstupnej matice, F je veľkosť filtra, P je veľkosť okraja a S je veľkosť posunu – kroku filtra. N, W, F, P aj S sú veľkosti udávané ako počet stĺpcov a počet riadkov.

Operáciu konvolúcie je možné vykonať aj opačným spôsobom, tzv. de-konvolúciou. Používa sa na prispôsobenie matice na pôvodnú veľkosť po konverzii tak, aby výstup CNN mal rovnaké rozmery ako vstup. Tento princíp sa využíva napríklad pri segmentácii obrázkov auto-enkóderom.

Častokrát sa za konvolučnými vrstvami nachádzajú aj vrstvy združovania – tzv. pooling. Združovanie zmenšuje rozmer aktivačných máp pri súčasnom zachovaní prieplustnosti

signálu. Existuje niekoľko typov združovania: maximálny, priemerný, summarizačný a ďalšie. V prípade maximálneho združovania (max pooling) je najprv definovaná veľkosť okna. Následne okno sa posúva obrazom ako filter v konvolúcii s tým rozdielom, že sa nasledujúce okno neprekryva s predchádzajúcim a výstup z okna je jeho najväčší prvok. V prípade priemerného združovania je výstupom priemer všetkých prvkov okna atď.



Obr. 21 Max pooling

Po použití niekoľkých konvolúcii a združení sa matica spracuje pomocou viacvrstvového perceptrónu MLP (Multi-Layer Perceptron). Úlohou MLP je použitie vlastností získaných z predchádzajúcich operácií na rozdelenie vstupov CNN do viacerých tried (segmentov).

Rovnako ako ANN, CNN tiež pozostávajú z vrstiev (konvolučných, združujúcich, aktivačných a MLP). Pri vytváraní CNN sa najčastejšie za sebou vyskytuje niekoľko blokov zložených z konvolučných a združovacích vrstiev. Tento vzor sa opakuje, kým sa vstupný obraz nerozdelí na malé kúsky. Medzi vrstvami môžu byť tiež paralelné vetvy. Nakoniec nasleduje vrstva MLP.

2.4 Rekurentné neurónové siete

Opakujúce sa neurónové siete alebo RNN sú rodinou neurónových sietí na spracovanie sekvenčných údajov. Tak ako konvolučná sieť je neurónová sieť, ktorá sa špecializuje na spracovanie mriežky hodnôt X , ako je napríklad obraz, opakujúca sa neurónová sieť je neurónová sieť, ktorá sa špecializuje na spracovanie sekvencie hodnôt $x^{(1)}, \dots, x^{(\tau)}$. Rovnako, ako sa konvolučné siete dajú ľahko škálovať na obrázky s veľkou šírkou a výškou

a niektoré konvolučné siete môžu spracovávať obrázky s premenlivou veľkosťou, rekurentné siete sa môžu škálovať na oveľa dlhšie sekvencie, ako by to bolo praktické pre siete bez špecializácie založenej na sekvenciách. Väčšina opakujúcich sa sietí môže spracovávať aj sekvencie rôznej dĺžky.

Zdieľanie parametrov umožňuje rozšíriť a použiť model na príklady rôznych foriem (napr. rôzne dĺžky) a zovšeobecniť ich. Keby sme mali samostatné parametre pre každú hodnotu časového indexu, nemohli by sme zovšeobecňovať na dĺžky sekvencií, ktoré sa počas tréningu nevideli, ani zdieľať získané vlastnosti na rôznych dĺžkach sekvencií a na rôznych pozíciah v čase. Takéto zdieľanie je zvlášť dôležité, keď sa špecifická informácia môže vyskytnúť na viacerých pozíciah v sekvencii.

Súvisiaca myšlienka je použitie konvolúcie naprieč 1-D časovou sekvenciou. Tento prístup je základom neurónových sietí s časovým oneskorením. Konvolúcia umožňuje sieti zdieľať parametre v čase, je však vágna. Výstupom je postupnosť, kde každý člen výstupu je funkciou malého počtu susedných členov vstupu. Myšlienka zdieľania parametrov sa prejavuje v aplikácii toho istého konvolučného jadra v každom časovom kroku. Rekurentné siete zdieľajú parametre iným spôsobom. Každý člen výstupu je funkciou predchádzajúcich členov výstupu. Každý člen výstupu je produkovaný pomocou rovnakého pravidla aktualizácie, aké sa použilo na predchádzajúce výstupy. Výsledkom tejto opakujúcej sa formulácie je zdieľanie parametrov prostredníctvom veľmi hlbokého výpočtového grafu.

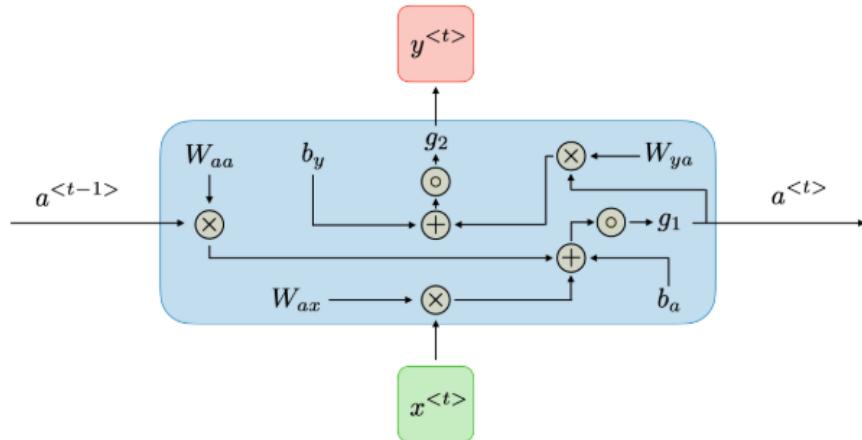
Kvôli jednoduchosti expozície označujeme RNN ako operujúce na sekvencii, ktorá obsahuje vektory $x^{(t)}$ s indexom časových krov t v rozsahu od 1 do τ . RNN môžu byť tiež aplikované v dvoch rozmeroch na priestorové dátu, ako sú obrázky, a dokonca aj keď sa aplikujú na dátu zahŕňajúce čas, siet môže mať spojenia, ktoré sa pohybujú dozadu v čase, za predpokladu, že je celá sekvencia pozorovaná skôr, ako je poskytnutá do siete [16].

Pre každý časový krok t , je hodnota aktivačnej funkcie $a^{(t)}$ a výstupu $y^{(t)}$ počítaná ako:

$$a^{(t)} = g_1(W_{aa}a^{(t-1)} + W_{ax}x^{(t)} + b_a) \quad (8)$$

$$y^{(t)} = g_2(W_{ya}a^t + b_y) \quad (9)$$

kde $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ sú parametre, ktoré sú zdieľané dočasne a g_1, g_2 sú aktivačné funkcie.



Obr. 22 Grafické znázornenie výpočtu aktivačných funkcií g_1 a g_2 a ich vplyvu na výstup a časové kontinuum [24]

Výhodou RNN sú vlastnosti ako možnosť spracovania vstupu akejkoľvek dĺžky, nemennosť veľkosti modelu vzhľadom na veľkosť vstupu, či zdieľanie váh naprieč časom. Na druhej strane, za nevýhodu týchto sietí sa dá považovať časovo náročné výpočtové spracovanie, či ťažkosti s prístupom informácií z dátnejšej minulosti.

Rekurentné neurónové siete je možné rozdeliť do 5 kategórií podľa vzťahu počtu vstupov a výstupov neurónovej siete [24]:

- **1-1 ($T_x = T_y = 1$)** – jeden vstup a jeden výstup, ide o tradičnú neurónovú sieť
- **1-n ($T_x = 1, T_y > 1$)** – jeden vstup a viacero výstupov, napr. generovanie hudby
- **n-1 ($T_x > 1, T_y = 1$)** – viacero vstupov a jeden výstup, napr. klasifikovanie sentimentu
- **n-n ($T_x = T_y$)** – viacero vstupov a viacero výstupov, pričom ich počet je rovnaký, napr. identifikovanie a kategorizovanie entít ako sú slová vo vete
- **n-n ($T_x \neq T_y$)** – viacero vstupov a viacero výstupov, pričom ich počet nie je rovnaký, napr. prekladanie textu z jedného jazyka do druhého.

2.5 Konfigurácia umelých neurónových sietí

Úspešné uplatňovanie techník hlbokého učenia si vyžaduje nielen dobrú znalosť toho, ktoré algoritmy existujú a princípov, ktoré vysvetľujú, ako fungujú. Podstatné je, ako si zvoliť algoritmus pre konkrétnu aplikáciu a ako monitorovať a reagovať na spätnú väzbu získanú z experimentov, aby sa zlepšil systém strojového učenia. Počas každodenného vývoja

systémov strojového učenia sa odborníci musia rozhodnúť, či zhromaždia viac údajov, zvýšia alebo znížia kapacitu modelu, pridajú alebo odstránia regularizačné funkcie, zlepšia optimalizáciu modelu, zlepšia približný odhad modelu alebo odladia jeho softvérovú implementáciu. Celkový proces by sa dal zhrnúť do nasledujúcich krokov [16]:

- **Určenie cieľov**, akú chybovú metriku použiť a cieľovú hodnotu pre túto chybovú metriku si zvoliť. Tieto ciele a metriky chýb by mali vychádzať z problému, ktorý má aplikácia vyriešiť.
- **Sfunkčnenie pracovného procesu od preprocesovania vstupných dát, cez natrénovanie modelu, postprocesovania predikcií až po odhad vhodných metrík na určenie úspešnosti modelu čo najrýchlejšie.**
- **Identifikovanie slabých miest.** Diagnostika komponentov, ktoré majú horšiu kvalitu, ako sa očakávalo, a či je to v dôsledku pretrénovania, pod-trénovania alebo chyby vo vstupných dátach alebo softvéri.
- **Opaková zmena** akou je získavanie nových dát, úprava hyperparametrov či zmena algoritmov vzhľadom na získané výsledky.

Úlohou učiaceho sa procesu je čo najpresnejšie riešiť daný problém, pričom sa častokrát ukáže získanie presnejšieho riešenia ako samotným človekom, čo nie je ojedinelý prípad. Pri vylepšovaní navrhnutého riešenia je potrebné sa sústrediť na zlepšovanie efektívnosti predikcií nie len vzhľadom k trénovacím vzorkám, ale predovšetkým k validačným vzorkám, na ktorých model neboli trénované. V prípade, že má model nízku úspešnosť na trénovacej vzorke, hovoríme o ňom, že je pod-trénovaný, zatiaľ čo v prípade že úspešnosť medzi predikciami na trénovacej vzorke a validačnej vzorke je príliš odlišná, hovoríme o pretrénovaní. Tieto dva fenomény sa pri hlbokom učení snažíme minimalizovať.

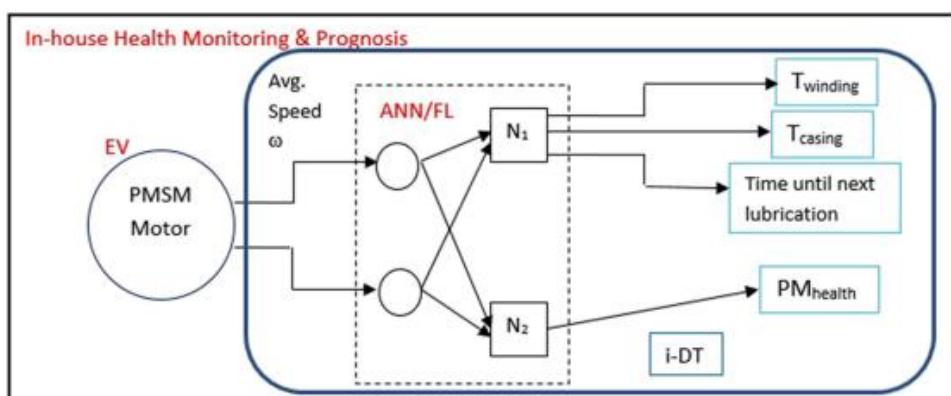
3 Inteligentná diagnostika v priemysle

V tretej kapitole dizertačnej práce sa venujeme prehľadu aktuálnej výskumnej činnosti v oblasti inteligentnej diagnostiky v priemysle. Skúmame vybrané vedecké články za účelom oboznámenia sa s aktuálnymi výzvami spojenými s inteligentnou diagnostikou v priemysle, vytvárame prehľad aktuálnych prístupov k riešeniam týchto výziev a hľadáme nedostatky a možnosti zlepšenia v danej oblasti.

Za posledných niekoľko rokov sa princíp monitorovania a riadenia mechatronických systémov vyvíjal prudko vpred, rovnako ako aj technologické vylepšenia ako také. V rámci analýzy súčasného stavu sme sa venovali predovšetkým vedeckým výskumom v oblasti inteligentného monitorovania a vyhodnocovania rôznych stavov robotických systémov, na základe ktorých by bolo možné vyhodnotiť chybovosť systému a tak systém zastaviť, či upozorniť na vyslanie technika k oprave stroja včas. Jednotlivým riešeniam sa venujeme podľa spôsobu ich realizácie a algoritmov, na ktorých sú postavené.

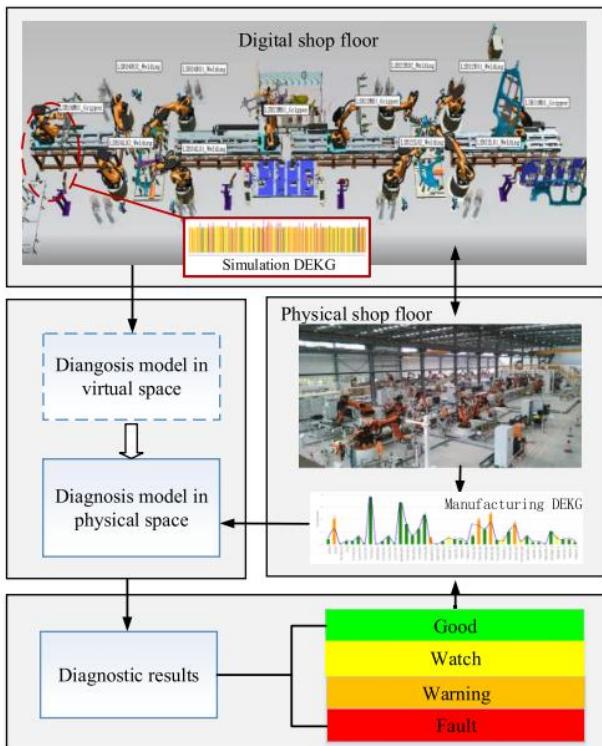
Robotické a kybernetické systémy sa stávajú čoraz zložitejšími a automatické identifikovanie komplexnejším. Preto sa aktuálne smerovanie vyhodnocovania stavu a zdravia systému vykonáva na báze umelej inteligencie a to predovšetkým na báze umelých neurónových sietí. Pojem „zdravie“ pre technické zariadenia je ekvivalentom pre „stav“ zariadenia.

Príkladom použitia umelých neurónových sietí vo svojej základnej podobe bez hlbokého učenia je štúdia na monitorovanie zdravia a prognóza elektrického motora za použitia inteligentného digitálneho dvojčaťa [25]. V tomto ponímaní použitia digitálneho dvojčaťa nejde o 3D vizualizáciu, ale o odzrkadlenie a premietnutie zdravia elektrického motora.



Obr. 23 Predikcia zdravia motora pomocou dvoch ANN [25]

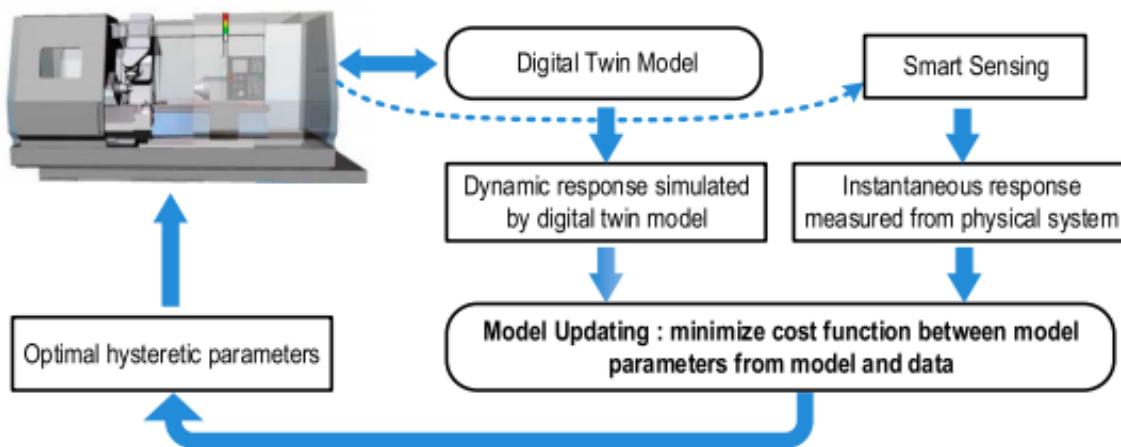
Dáta potrebné na natrénovanie siete autori získali na základe dát z reálneho prístroja kedy sledovali rýchlosť akou sa vozidlo pohybovalo a čas, definujúci koľko bol motor zapnutý. Výsledkom na odzrkadlenie zdravia motora boli parametre ako teplota plášťa motora, teplota statorového vinutia, čas na výmenu ložiskového maziva a samotné zdravie rotora. Zdravie rotora je pritom definované ako percentuálne zhorsenie a odhad zostávajúcej užitočnej životnosti. V navrhnutom riešení využili autori dve jednoduché neurónové siete.



Obr. 24 Použitie pred-trénovaného modelu pre diagnostiku zariadenia [26]

Štúdia [26] sa zaoberá diagnózou chybovosti za využitia digitálneho dvojča a hlbokého učenia v podobe pred-trénovania modelu na umelých dátach získaných z digitálneho dvojča a následne dotrénovaného na reálnom zariadení. V tomto prípade nejde o skutočné digitálne dvojča, ale skôr len o digitálny tieň, kedy je model zariadenia plne virtuálny a nie je nijak prepojený s fyzickým modelom. Ako model neurónových sietí štúdia využila SSAE (Stacked Sparse Auto-Encoder) model, čo predstavuje architektúru pozostávajúcu z niekolkých na seba nadväzných auto-enkóderov s mnohými neaktívnymi skrytými uzlami. Na základe pred-trénovaného modelu na virtuálnych dátach je model ďalej dotrénovaný distribúciou dát z reálneho prostredia na základe elektrokardiogramu zariadenia (DEKG).

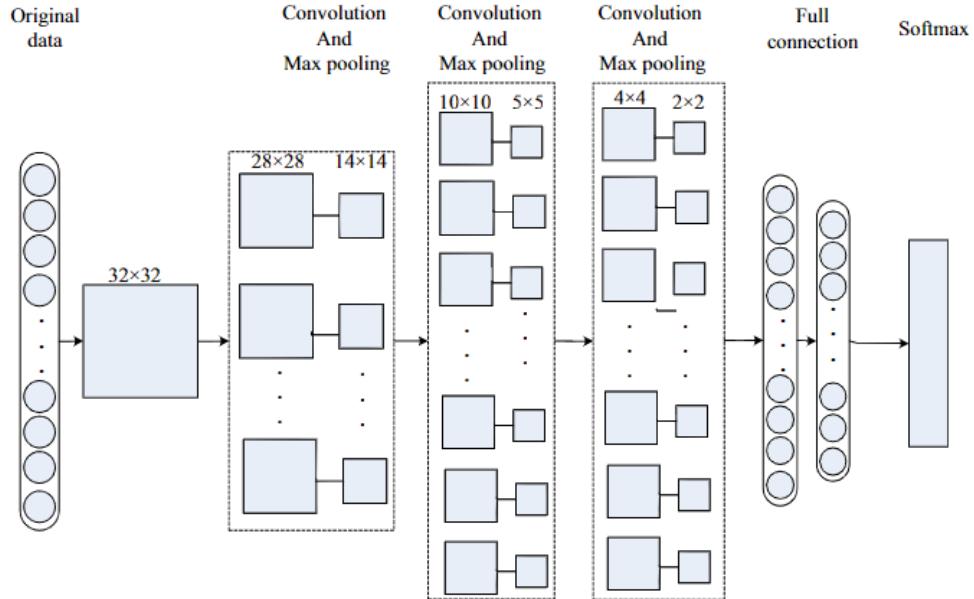
Digitálnym dvojčaťom pre odhalenie chybovosti, monitorovania zdravia zariadenia a riadenia systému sa inšpirovala aj štúdia aplikovaná na rotačné stroje [27]. V tomto prípade sa výskumníci zamerali na monitorovanie a riadenie rotačných strojov, pričom na rozhodovanie využili dátu z digitálneho dvojčaťa. V tomto prípade sa dá povedať, že ide o digitálne dvojča, ktoré s reálnym zariadením je späť na úrovni riadenia a porovnávania stavov. Skutočné zariadenie sa optimalizuje na základe porovnania odsimulovaných dát digitálneho modelu a dát získaných z reálneho zariadenia a následne sa digitálny model aktualizuje o aktuálny stav zariadenia, čo má za následok zrkadlenie modelu a zariadenia pričom sú obe inštancie regulované aby sa synchronizovali. V konečnom dôsledku ide o kalibráciu digitálneho modelu vzhľadom k reálnemu zariadeniu, aby čo najpresnejšie odzrkadl'oval skutočné zariadenie.



Obr. 25 Kalibrácia digitálneho modelu na základe odozvy reálneho zariadenia [27]

V posledných piatich rokoch sa využívali neurónové siete na diagnostikovanie chyby zariadenia s rôznorodými architektúrami. Boli použité jednoduché neurónové siete ako napríklad v štúdii [25] alebo čoraz viac využívané hlboké neurónové siete ako v štúdii [26]. Úspešnosť hlbokých neurónových sietí je vyššia než jednoduchých neurónových sietí či matematických modelov a s narastajúcim objemom dát sa oproti ostatným riešeniam úspešnosť nadálej zvyšuje. Od tejto skutočnosti sa odvíjali aj mnohé príspevky vo vedeckých časopisoch či konferenciách a na klasifikáciu stavu zariadenia využili hlboké neurónové siete. Nakoľko sa stav zariadenia mení v čase a je od neho závislý, je možné na klasifikáciu stavu zariadenia využiť aj konvolučné aj rekurentné siete. V prípade

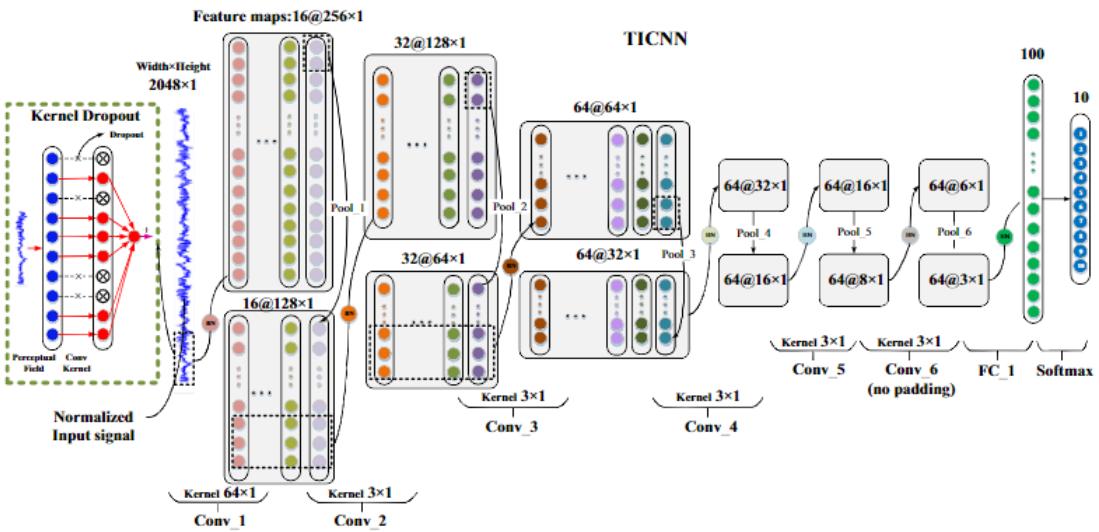
klasifikovania segmentov stavu zariadenia je možné použiť hlbokú neurónovú sieť ako v prípade [28], [29], [30] alebo [31], kde boli použité konvolučné neurónové siete.



Obr. 26 ADCNN architektúra [28]

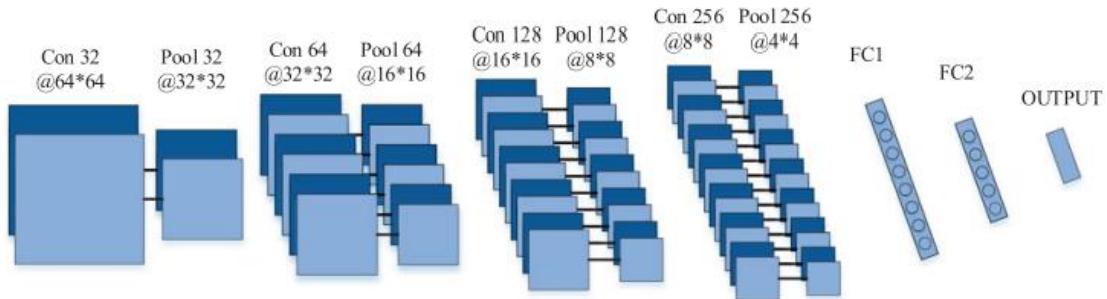
Štúdia zaoberajúca sa odhalením a zaradením chyby na ložisku [28] využila na diagnostikovanie zariadenia adaptívnu hlbokú konvolučnú neurónovú sieť. Architektúru tejto siete v procese rozpoznania chyby a jej závažnosti využili v prvom kole na definovanie chyby, či sa vyskytuje na vonkajšej drážke, vnútorej drážke, v ložiskovej guli alebo je zariadenie v poriadku. Po zaklasifikovaní stavu zariadenia do jednej zo spomenutých kategórií bola sieť opäť použitá, no v tomto prípade na určenie veľkosti chyby v palcoch.

Ďalšou štúdiou, ktorá sa zaoberala odhalením chyby na ložisku bola [29], kde základ tvorila neurónová sieť zostavená z konvolučných blokov za pomocí inferencie počas tréningu. V tomto prípade slúži inferencia na odstránenie šumu signálu predtým, ako je na ňom sietť trénovaná. Sietť je trénovaná na signáloch vibrácií a tým pádom využíva konvolučné neurónové siete v tzv. 1D móde. Konvolučné siete sú navrhnuté tak, aby operovali s 2D dátami na vstupe, avšak môžu byť upravené aj tak, aby operovali s 1D dátami. Práve v takýchto prípadoch sa využívajú konvolučné siete na klasifikáciu signálu a nie len obrazu.



Obr. 27 TICNN architektúra [29]

Na diagnostiku rotačných strojov sa zamerala štúdia [30], v ktorej bolo zdravie prístroja definované na základe zachytených vibračných signálov. Sieť v tomto prípade mala jednoduchú architektúru pozostávajúcu z jednej konvolučnej vrstvy s počtom filtrov 32 o veľkosti 2x64. Následne za konvolučnou vrstvou nasledovala plne prepojená vrstva o veľkosti 200 uzlov a nato už samotná klasifikácia sledovaného signálu.

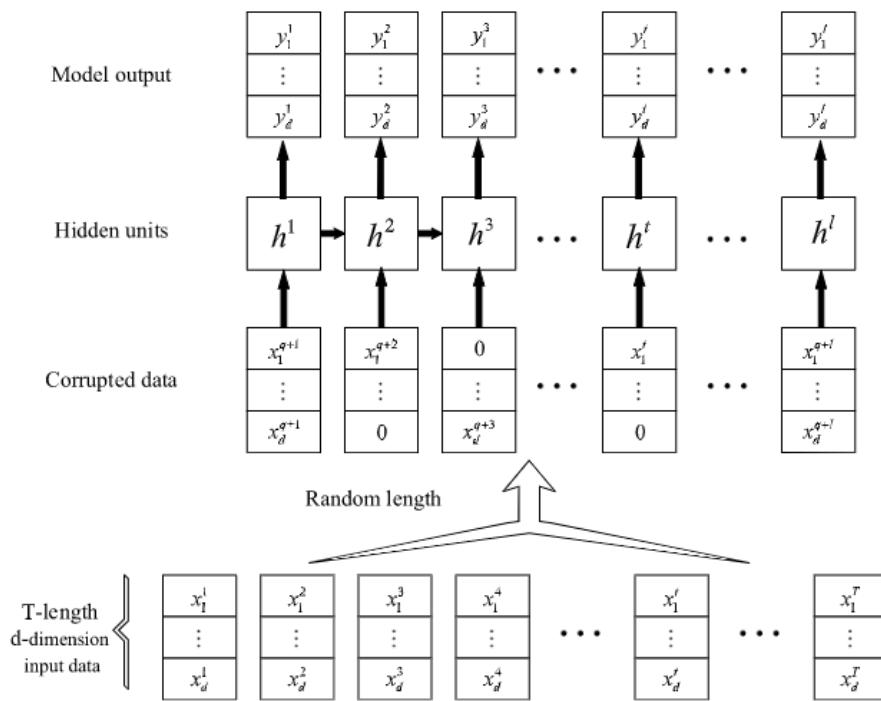


Obr. 28 Klasifikačná CNN sieť pre vstupy o veľkosti 64x64 [31]

Možnosť konverzie signálu na obrazovú štruktúru predstavujúcu časový horizont využila štúdia [31]. Po konverzii zachyteného signálu do obrazového formátu bola na diagnostiku využitá architektúra konvolučných neurónových sietí s názvom LeNet-5. Východisková veľkosť vstupu tejto architektúry je 2D vstup o veľkosti 32x32, no v rámci výskumu boli testované aj veľkosti 16x16 a 64x64. Pre trénovanie a testovanie navrhnutej metódy boli

použité signály vibrácií z motorových ložísk, signály vibrácií samo-nasávacieho odstredivého čerpadla a signály vibrácií axiálneho piestového hydraulického čerpadla. Samotná architektúra siete pozostáva zo 4 za sebou idúcich blokov. Tieto bloky predstavujú konvolučné vrstvy a združovacie (pooling) vrstvy za nimi na zníženie dimenzionality a výpočtovej náročnosti. Za týmito blokmi následne idú dve po sebe idúce plne prepojené vrstvy zakončené samotným klasifikačným výstupom. Túto architektúru predstavuje Obr. 28.

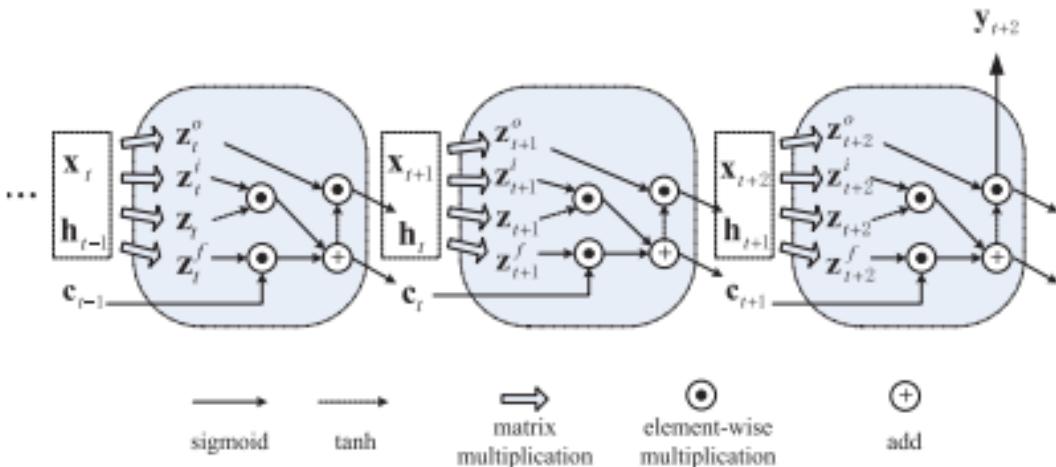
Inou možnosťou je využitie rekurentných neurónových sietí ako v prípade článku [32] a [33]. Aj napriek tomu, že práve rekurentné siete sú orientované na časovo-orientované dátá, tak dopyt po nich nie je natol'ko rozsiahly. Dôvodom môže byť vysoká časová náročnosť na výpočty počas tréningu sietí.



Obr. 29 Architektúra GRU-NP-DAE [32]

Obr. 29 zobrazuje využitie ako rekurentnej siete, tak aj auto-enkódera v rámci jednej architektúry, ktorú navrhla štúdia [32] v rámci diagnostikovania stavu zariadenia pre odhalenie chyby ložiska rotačného stroja pomocou signálov emitovaných vibrácií. V rámci článku bola architektúra využitá ako GRU-NP-DAE. GRU značí „Gated Recurrent Unit“, čiže bránovú rekurentnú jednotku, NP je označenie pre „Non-linear Predictive“, čiže

nelineárny prediktívny, zatialčo DAE hovorí o „Denoising Auto-Encoder“ – šum odstraňujúci auto-enkóder. Preto, o modeli GRU-NP-DAE by sa dalo hovoriť ako o nelineárnom prediktívnom šum odstraňujúcom auto-enkóderi postavenom na bránových rekurentných jednotkách. Vstupnými dátami pre tento model sú signály vibrácií otočného stroja s vysokým podielom zašumenia z okolia. Zašumený signál je prevádzaný pomocou šum odstraňujúceho auto-enkódera na extrahované vlastnosti, ktoré sú ďalej spracovávané cez bránové rekurentné jednotky. Výstupom tohto modelu sú predpovedané hodnoty signálu s preddefinovaným omeškaním (využitie nelineárneho prediktívneho auto-enkódera). Týmto princípom sú natrenované auto-enkódery pre každú zo špecifikovaných chýb, ako aj pre stav za normálneho behu stroja. V rámci diagnostiky zariadenia sú jednotlivé časové úseky predikované každým z natrenovaných modelov a na základe najmenšej odchýlky reálnych hodnôt je napokon kategória signálu vibrácií vyhodnotená buď ako bezchybný chod zariadenia alebo jedna z definovaných chýb.

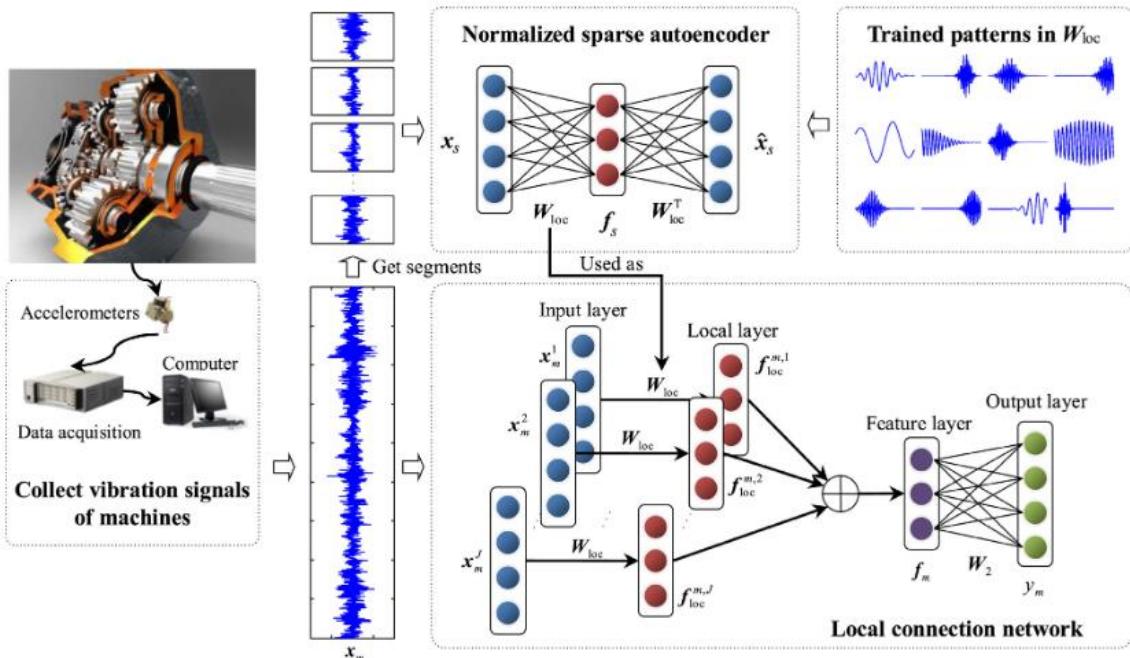


Obr. 30 Vanilla LSTM [33]

Príkladom použitia rekurentných sietí pre diagnostikovanie stavu je aj štúdia [33], v ktorej sa venujú diagnostikovaniu chýb chemických procesov. Na diagnostiku využili architektúru „vanilla LSTM“. Vanilla LSTM je architektúra, ktorá má reťaz spojení ako klasická RNN, avšak opakovacia jednotka je odlišná. Namiesto 1 vrstvy sa tu nachádzajú vrstvy 4, ktoré medzi sebou interagujú v špeciálne definovanej štruktúre, ktorú predstavuje Obr. 30. Vnútorná štruktúra vanilla LSTM je založená na súbore prepojených buniek.

Rozdielne sú v tom, že klasické RNN prepisujú stav bunky priamo, zatiaľ čo vanilla LSTM má bunky, ktoré obsahujú 3 brány slúžiace ako riadiaca jednotka šírenie informácie v sieti.

Klasickým prístupom k riešeniu problému odhalenia chyby na prístroji sa javí využitie architektúry auto-enkódera, ktorému sa venujú štúdie [26], [32], [34], [35] a [36]. V rámci výskumu [34] bol použitý SAE – „Sparse Auto-Encoder“ – riedky auto-enkóder. Pod riedkym auto-enkóderom si predstavujeme taký auto-enkóder, ktorý nemá všetky prepojenia medzi susediacimi vrstvami prepojené, a tým nadobúda vlastnosť „riedky“. V rámci tohto výskumu bolo natrénovaných niekoľko auto-enkóderov, z ktorých sa vyňala zakódovaná časť, ktorá predstavovala daný stav systému a využila sa pri zostavení klasifikačnej architektúry vykreslenej na Obr. 31.



Obr. 31 Architektúra využívajúca zakódované vlastnosti z auto-enkóderov pri klasifikácii stavu zariadenia. [34]

V rámci článku [35] sa autori zaoberali využitím SAE pre diagnostiku indukčných motorov. Tak ako v predchádzajúcim článku, tak aj autori tohto článku sa rozhodli využiť auto-enkóder na získavanie vlastností signálov. V tomto prípade šlo o šum odstraňujúci auto-enkóder, ktorého vlastnosti sa využili na prečistovanie signálu a tak jeho jednoduchšiu klasifikáciu. Rozdielom použitia auto-enkódera ako vlastnosti získavajúcej architektúry pre každú z definovaných chýb ako aj optimálneho stavu nebola ani štúdia [36] na diagnostikovanie signálov vibrácií rotačného stroja. Rozdielny bol spôsob optimalizácie

parametrov auto-enkódera počas trénovacieho procesu. Na rozdiel od typických optimalizačných algoritmov neurónových sietí akými sú napr. ADAM alebo SGD bol použitý algoritmus AFSA. Netradičným elementom bola aj samotná chybová funkcia. Častým použitím je MSE – stredná kvadratická chyba, avšak autori sa rozhodli použiť maximálnu korrentropiu. Maximálna korrentropia sa v posledných rokoch využíva predovšetkým ako chybová funkcia na spracovávanie zašumiených signálov neurónovými sietami.

Po podrobnej analýze vybraných vedeckých prác zachytávajúcich prístupy k inteligentnej diagnostike zariadení je možné sledovať nasledovné trendy:

- Využívanie hlbokého učenia.
- Využívanie predovšetkým konvolučných neurónových sietí.
- Sledovanie zmien sledovanej veličiny opisujúcej stav zariadenia v čase.
- Získavanie vlastností sledovanej veličiny definujúcich jednotlivé stavy zariadenia.
- Využívanie auto-enkóderov na extrahovanie vlastností.

Prehľad výskumnej činnosti definovanej vybranými vedeckými prácami poukazuje na nedostatok daných prác v jednom alebo vo viacerých bodoch:

- Prispôsobivosť navrhnutého algoritmu na iný problém podobnej triedy nie je definovaný.
- Vytváranie modelov neurónových sietí na extrakciu vlastností je pre každý stav osobitne, čím sa zväčšuje časová a výpočtová náročnosť inteligentných algoritmov.
- Definovanie stavu zariadenia je na základe iba jednej sledovanej veličiny, čím sa zužuje spektrum odhaliteľných chýb.
- Definovanie stavu zariadenia je na základe celých časových blokov bez opakovanej predikcie v určitých bodoch, čo má za následok oneskorenie odhalenia prípadnej chyby.

4 Ciele dizertačnej práce

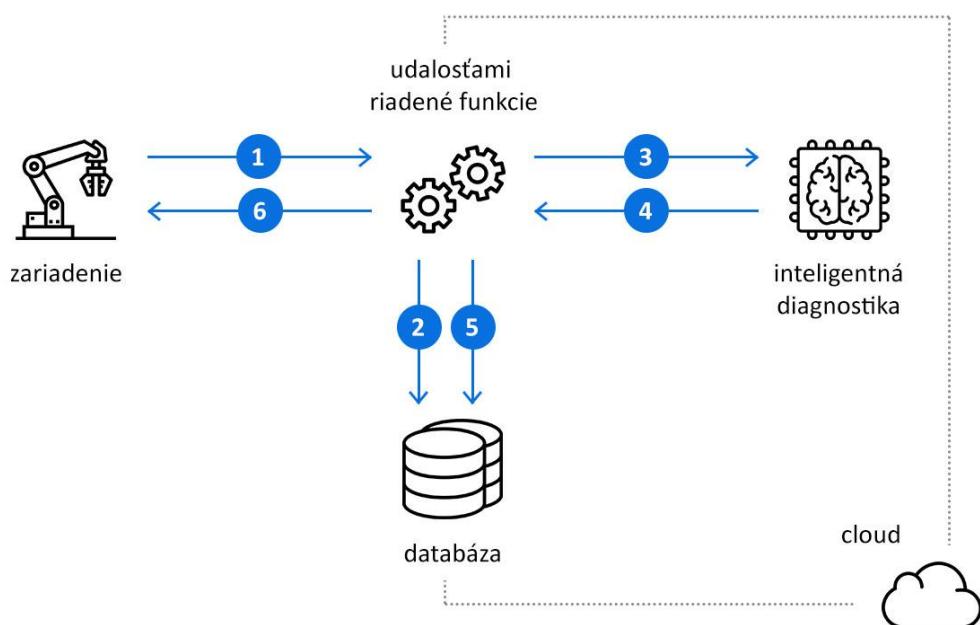
V štvrtej kapitole dizertačnej práce definujeme ciele dizertačnej práce, ktoré vyplývajú z naštudovanej literatúry a naštudovanej vybranej výskumnnej činnosti v danej oblasti, ktoré sú opísané v prvej až tretej kapitole dizertačnej práce.

Monitorovanie a riadenie mechatronických systémov v priemysle sú jeho neoddeliteľnou súčasťou a postupom času sa neustále zdokonalujú. Monitorovanie takýchto zariadení je podstatnou zložkou životného cyklu mechatronických systémov, pretože od výsledku monitorovania sa odvádzajú aj ďalšie kroky ich životného cyklu akými sú údržba, riadenie či optimalizácia. Tieto technológie sú neustále rozvíjajúci sa a vyvíjajúci sa komplex, v ktorom je vždy priestor na rozvoj. Aktuálna, neustále napredujúca paradigma Industry 4.0. spája najnovšie informačno-komunikačné technológie akými sú IoT, Big Data, Cloudové výpočty, automatizované procesy či umelá inteligencia. Na základe analýzy dostupnej literatúry a výskumných projektov v oblasti diagnostiky chodu zariadenia v predošlej kapitole možno konštatovať, že inteligentná diagnostika zdravia zariadení a procesov sa zameriava predovšetkým na konkrétné zariadenia a procesy, jednu sledovanú systém opisujúcu hodnotu meniacu sa v čase, pričom sledovaná hodnota je vyhodnocovaná pre celý časový úsek ako jednotná, bez možnosti sledovania menších odchýlok, ktoré by mohli včas detegovať aj menšie výkyvy v správaní.

Požiadavky rozvíjajúceho sa priemyslu neustále rastú v každom smere, aby bola jeho expanzia zabezpečená s čo najhladším priebehom, ktorý zabezpečí čo najnižšie náklady pre jeho chod, ako aj pre aplikáciu zmien plánovaných ale aj neplánovaných. V priemyselnej oblasti jednou zo základných zložiek zabezpečujúcich znižovanie nákladov je udržiavanie dobrého zdravia zariadenia. Včasné odhalenie chyby zariadenia, prípadné odchýlky od štandardného pracovného režimu je kľúčovým aspektom pri plánovaní jeho prípadnej údržby s čo najmenším dopadom na výrobný proces, ako aj skoré zareagovanie na kritické stavy eliminujúce úplné zlyhanie systému.

Výskum a vývoj včasnej a efektívnej diagnostiky sledovaného systému v sietiach IIoT predstavuje multidisciplinárnu úlohu, na ktorú prepája sledovanie a riadenie mechatronického zariadenia, komunikáciu zariadenia s decentralizovanou výpočtovou jednotkou, spracovanie prichádzajúcich informácií, umelej inteligencie na predikciu zdravia a prípadnej chyby zariadenia s včasným zareagovaním na kritické stavy inštruovaním

sledovaného systému na jeho zastavenie, aby sa predišlo neodvratiteľným poškodeniam spôsobených zlyhaním zariadenia.



Obr. 32 Základný návrh intelligentnej diagnostiky a riadenia daného zariadenia

Na základe analýzy aktuálneho stavu výskumu diagnostiky mechatronických zariadení ako aj potrieb trhu, sme navrhli koncept kontinuálneho sledovania zariadenia opisovaný viacerými sledovanými premennými, jeho diagnostikovania v každom bode sledovaného času a prípadné zareagovanie v pretrvávajúcom kritickom stave, ktorý zobrazuje Obr. 32. Tok informácií návrhu intelligentnej diagnostiky pozostáva z nasledujúcich bodov tak, ako sú naznačené v obrázku:

1. Odoslanie sledovaných premenných opisujúci chod zariadenia do cloutu na výpočtovú jednotku
2. Uloženie získaných informácií do databázy
3. Odoslanie najnovších informácií o zariadení v preddefinovanom časovom horizonte umelej inteligencii na predikciu stavu zariadenia
4. Odoslanie vyhodnotenia predikcie stavu zariadenia umelou inteligenciou na výpočtovú jednotku
5. Uloženie predikcie stavu zariadenia
6. V prípade pretrvávajúceho kritického stavu, inštruovanie zariadenia na jeho zastavenie výpočtovou jednotkou.

Na základe analýzy súčasného stavu v študovanej oblasti sme stanovili nasledujúce základne ciele dizertačnej práce:

1. Návrh moderných metód monitorovania a riadenia mechatronických systémov na báze umelej inteligencie pre diagnostické postupy.
 - a. Definovanie požiadaviek na výsledný systém.
 - b. Koncept vlastného UI systému s využitím súčasných postupov sledovania a ovládania mechatronických systémov.
2. Implementácia moderných metód monitorovania a riadenia mechatronických systémov s dôrazom na diagnostiku zariadenia vzhľadom na monitorované veličiny za využitia najmodernejších prístupov.
 - a. Výber sledovaných mechatronických zariadení.
 - b. Inteligentná diagnostika zvolených systémov.
 - c. Komunikácia vybraných zariadení so systémom inteligentného monitorovania a riadenia.
 - d. Ukladanie dát o zariadeniach a ich riadenie na základe analýzy ich stavu.
3. Overenie navrhnutého riešenia diagnostiky mechatronických systémov na vybraných zariadeniach.
4. Zhrnutie možností využitia novej metodiky a nových metód monitorovania a riadenia mechatronických zariadení pre ich využitie v priemyselnej praxi (Industry 4.0) v oblasti mechatronických systémov.

5 Návrh systému monitorovania a riadenia mechatronických systémov

V piatej kapitole dizertačnej práce sa venujeme vlastnému návrhu systému monitorovania a riadenia mechatronických systémov na základe stanovených cieľov dizertačnej práce. Definujeme funkcionálne aj nefunkcionálne požiadavky na výsledný systém. Vzhľadom na definované požiadavky navrhujeme koncept výsledného systému, ktorý spĺňa ciele práce.

Hlavným zámerom predloženej dizertačnej práce je návrh a realizácia kontinuálnej inteligentnej diagnostiky, jej vyhodnocovania a reagovania na zmeny sledovaného zariadenia na základe systému opisujúcich premenných meniacich sa v čase v súlade s konceptom Industry 4.0. tak, aby navrhnuté riešenie bolo aplikovateľné na širšie spektrum zariadení. Takto navrhnuté riešenie by malo spĺňať definované funkcionálne a nefunkcionálne požiadavky.

5.1 Požiadavky na výsledný systém

Výsledný návrh a realizácia riešenia vychádza zo špecifických požiadaviek, ktoré vyplývajú z analýzy aktuálneho stavu problematiky inteligentnej diagnostiky a riadenia mechatronických systémov. Jednotlivé požiadavky delíme na funkcionálne a nefunkcionálne požiadavky na výsledný systém.

5.1.1 Funkcionálne požiadavky na výsledný systém

- Zariadenie sa musí vedieť pripojiť na cloud cez internet.
- Zariadenie musí vedieť kontinuálne odosielat' dátu.
- Odoslané dátu musia obsahovať časovú pečiatku v UTC časovom pásme.
- Prijaté dátu na cloud musia byť okamžite po prijatí obslúžené.
- Prijaté dátu na cloud sa musia uložiť do databázy.
- Po uložení najnovších dát do databázy musí byť diagnostikovaný stav zariadenia vzhľadom na čo najaktuálnejšie dátu.
- Výsledok diagnostiky musí byť vyhodnotený pre každý zaznamenaný časový bod.
- Výsledok diagnostiky musí byť uložený v databáze.

- Viačnásobná predikcia jedného časového bodu musí byť vyhodnotená ako jeden zo stavov podľa vyššieho počtu zastúpení jednej triedy.

5.1.2 Nefunkcionálne požiadavky na výsledný systém

- Modulárnosť systému.
- Adekvátna rýchlosť získavania údajov.
- Adekvátna rýchlosť spracovania diagnostiky.
- Ľahké a dostupné prispôsobenie algoritmu diagnostiky pre zabezpečenie modulárnosti systému.
- Ľahká dostupnosť zdrojov.

5.2 Koncept výsledného systému

Koncept výsledného systému je rozdelený na tri hlavné úrovne a to úroveň zariadenia, úroveň clodu a úroveň prepojenia. Jednotlivé úrovne konceptu výsledného systému odzrkadľujú základné body toku dát od ich namerania až po vyhodnotenie.

V rámci základného návrhu riešenia z Obr. 32 je možné definovať koncept riešenia na hlavné úrovne aj s ich jednotlivými komponentami nasledovne:

- **Úroveň zariadenia**
 - Pripojenie na internet
 - Prijímanie a odosielanie dát
- **Úroveň clodu**
 - Udalostami riadené funkcie
 - Databáza uspôsobená pre Big Data analýzy
 - Umelá inteligencia
- **Úroveň prepojenia**
 - IoT
 - Pub-Sub komunikačný protokol

5.2.1 Úroveň zariadenia

Pre splnenie definovanej požiadavky ľahkej dostupnosti zdrojov boli modelové zariadenia pre testovanie navrhnutého riešenia vybrané ako zariadenia operujúce na mikrokontroléroch

– jednočipových mikropočítačoch akými sú napríklad dosky Arduino, Adafruit alebo Wemos.

AutomationShield je voľne otvorená iniciatíva zameraná na vytváranie mechatronických zariadení pre výučbu v oblasti riadenia a mechatroniky. Tieto zariadenia sa priamo pripájajú na programovateľné mikropočítače. Momentálne je podporovaných niekoľko typov Arduino architektúr a Adafruit Metro M4 express [37]. Pre účely dizertačnej práce boli vybrané dve laboratórne zariadenia – FloatShield – zariadenie vzduchom vznášajúcej loptičky v tube; a MotoShield – zariadenie na simulovanie spätej väzby rýchlosťi a pozície motora.

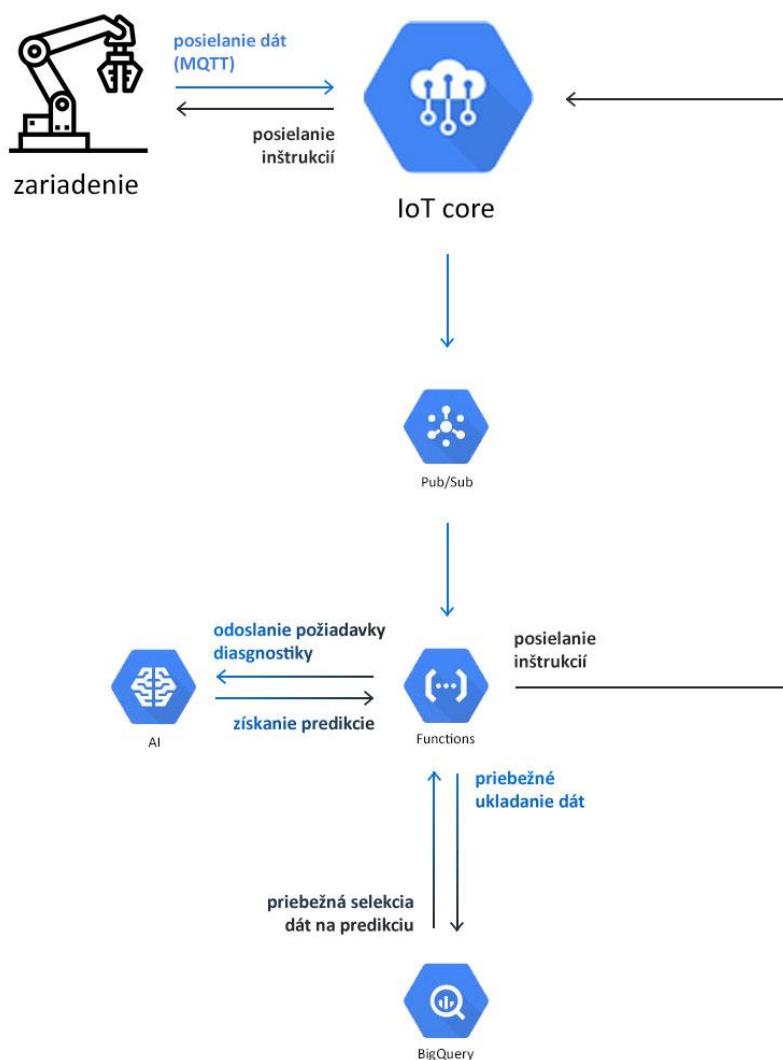
Tieto zariadenia boli umiestnené na mikropočítači Arduino Uno. Samotné Arduino Uno nepodporuje internetovú komunikáciu a preto bola použitá doska Uno R3+Wifi. Táto doska v sebe obsahuje dva čipy – ATmega328, na ktorom je postavené aj samotné Arduino Uno, a ESP8266, ktorý slúži na bezdrôtovú komunikáciu. Uno R3+Wifi doska má rovnaké rozmiestnenie základných PIN-ov ako Arduino Uno + má ďalšie pre Wifi čip. Tieto dva čipy môžu pracovať samostatne, alebo navzájom komunikovať cez sériovú zbernicu.

5.2.2 Úroveň cloutu

Na trhu je niekoľko dostupných cloudových riešení pre malé, stredné až veľké podniky. Medzi najznámejšie a najškálovateľnejšie patria Microsoft Azure, Amazon Web Services, Google Cloud Platform, IBM Cloud, Alibaba Cloud a Oracle Cloud. Potreby dizertačnej práce na úrovni cloutu a na úrovni prepojenia zariadenia s clodom je možné naplniť každým spomenutým poskytovateľom, okrem Oracle Cloud, ktorý momentálne neposkytuje možnosti umelej inteligencie. Ďalším parametrom výberu cloudového poskytovateľa je aplikovateľnosť a vhodnosť frameworku na tvorbu umelej inteligencie pomocou neurónových sietí. Medzi najznámejšie frameworky na tvorbu neurónových sietí patria TensorFlow, PyTorch, Keras, MXNet, Caffe, Theano a Microsoft CNTK. Vďaka týmto frameworkom sa dostávame do užšieho výberu cloudového poskytovateľa, nakoľko TensorFlow je najlepšie kombinovateľný s Google Cloud Platform, MXNet s Amazon Web Services a Microsoft CNTK s Microsoft Azure. Framework Keras je API, ktorý je možné postaviť na Theano, CNTK ale aj TensorFlow. Nakoľko je Tensorflow, ako knižnica pre tvorbu algoritmov hlbokého učenia, framework s najrozsiahlejšou komunitou, pričom je možné popri jej využívaní použiť aj Keras, tak bola práve táto knižnica spolu s Keras frameworkom zvolená pre splnenie potrieb dizertačnej práce v procese tvorby algoritmov

hlbokého učenia a inteligentnej diagnostiky zariadenia. Vzhľadom na výber knižnice bol vybraný aj najkompatibilnejší clouдовý poskytovateľ ako Google Cloud Platform.

5.2.3 Úroveň prepojenia



Obr. 33 Štruktúra prepojenia jednotlivých komponentov úrovní konceptu riešenia v prostredí Google Cloud Platform

Obr. 33 vyobrazuje podrobnejú štruktúru návrhu riešenia inteligentnej diagnostiky a riadenia zariadenia pomocou algoritmických výpočtov umiestnených na cloude – špecificky v prostredí Google Cloud Platform. Mechatronické zariadenie pripojené na internet komunikuje so službou IoT Core cez protokol MQTT. Toto zariadenie posiela správy vo

formáte JSON, ktorý obsahuje časovú pečiatku a údaje o zariadení v časovom bode. Služba IoT Core ďalej deleguje správy zo zariadenia na bez-serverové udalostami evokované funkcie Cloud functions. Tieto funkcie sa spúšťajú pri každom prijatí správy v kanáli Pub/Sub na konkrétnu tému, ktorá sa vzťahuje k zariadeniu. Prijaté dátá funkcia prekonvertuje a uloží do databázy BigQuery, ktorá slúži na analytické účely a je ju možné dopytovať pomocou SQL príkazov. Po uložení najnovších dát zo zariadenia si Cloud function zoberie najnovší segment dát, ktoré slúžia ako vstup pre algoritmy umelej inteligencie na predikciu stavu zariadenia. Tieto dátá ďalej deleguje na vytvorenú a natrénovanú neurónovú sieť. Táto neurónová sieť požiadavku spracuje a predikciu pošle naspäť na Cloud function. Cloud function následne aktualizuje tabuľku BigQuery o predikciu z neurónovej siete. Podľa charakteru stavu zariadenia Cloud function vyhodnotí, či je potrebné zariadenie predčasne zastaviť. Ak áno, IoT Core je priamo inštruovaný aby poslal príkaz na zariadenie na jeho zastavenie.

Ako výsledný koncept systému monitorovania a riadenia mechatronických systémov so zameraním na diagnostické postupy s využitím inteligentných algoritmov sme navrhli postupy v súlade s najnovšími trendmi v oblasti informačno-komunikačných technológií podľa paradigmy Industry 4.0. Mechatronické zariadenie komunikuje s decentralizovanou clouдовou výpočtovou jednotkou, ktorá pomocou udalostami riadenými funkciami prijíma, spracováva a vyhodnocuje dátá zo zariadenia. Tieto dátá ukladá do databázy, dopytuje inteligentné algoritmy na vyhodnotenie stavu zariadenia.

6 Implementácia systému monitorovania a riadenia mechatronických systémov

V šiestej kapitole dizertačnej práce sa venujeme implementácii autorkou navrhnutých algoritmov inteligentnej diagnostiky zariadení. Tieto algoritmy sú otestované na vybraných mechatronických zariadeniach a sú implementované v decentralizovanej výpočtovej jednotke clodu. Úlohou dizertačnej práce bolo navrhnuť komplexný systém pre monitorovanie a riadenie mechatronických systémov na báze umelej inteligencie pre diagnostické postupy, ktorý je možné ľahko implementovať v rámci jednoducho dostupných cloudových služieb. Po navrhnutí daného systému nasleduje samotná implementácia riešenia, ktorá sa opiera o návrh systému.

Na základe definovaných potrieb a požiadaviek na systém sme navrhli riešenie tak, ako je zobrazené na Obr. 32. V takomto prípade nie je potrebné zaobstarávať vlastný server, a to ani ako cloudovú službu, nakoľko sa využívajú tzv. bez-serverové služby. Je možné využívať jednotlivé komponenty ponúkané cloudovým poskytovateľom a prepojiť ich na úrovni týchto služieb, bez nutnosti programovania vlastného serverového rozhrania.

Samotná implementácia systému pozostáva z troch hlavných prepojených bodov:

- **Mechatronické zariadenie** – sledované zariadenie.
- **Inteligentná diagnostika** – neurónová siet' na vyhodnocovanie stavu zariadenia na základe aktuálneho časového úseku.
- **Cloudové služby** – služby pre prepojenie zariadenia s výpočtovo náročnou inteligentnou diagnostikou vyhodnocujúcou aktuálny stav zariadenia, ako aj priebežné ukladanie sledovaných hodnôt a analýz.

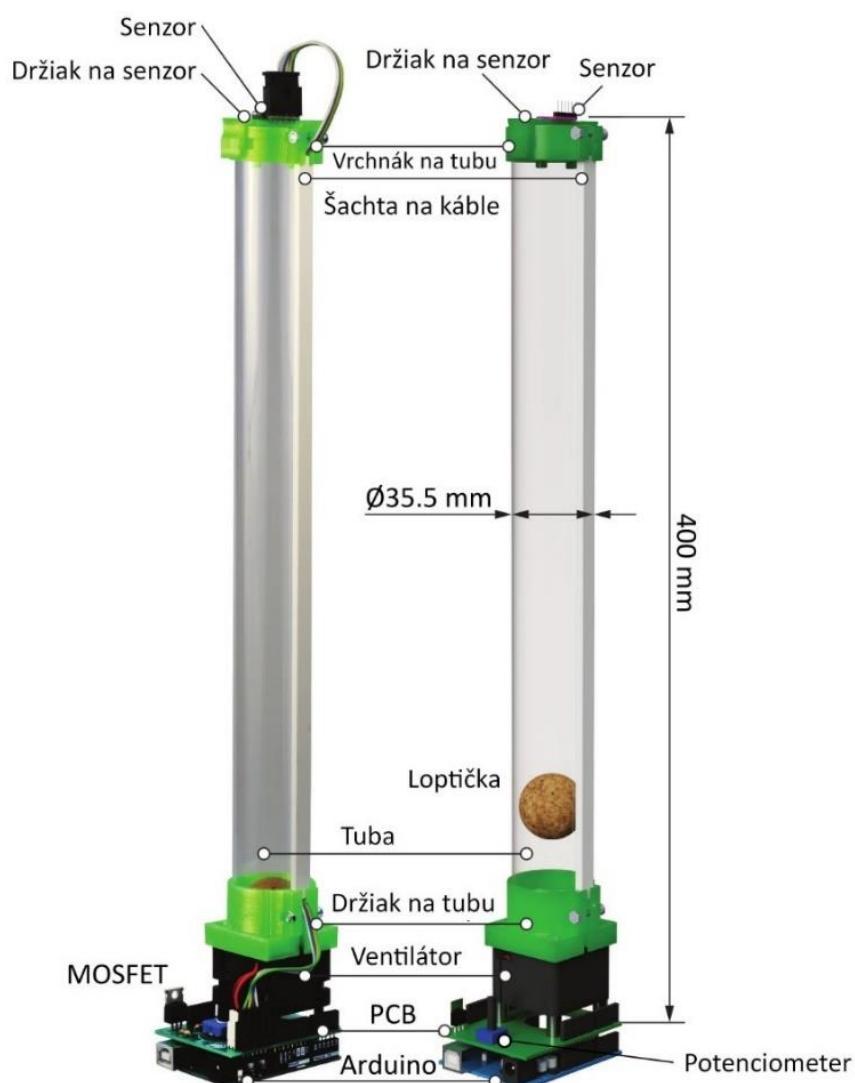
6.1 Mechatronické zariadenie

Ako ukážkové zariadenie boli vybrané dva laboratórne systémy: FloatShield a MotoShield z rodiny Arduino Shieldov AutomationShield pre výučbu systémov riadenia a mechatronických systémov [38] [37]. Jedná sa o voľne dostupné a lacné riešenie, ktoré si na základe dodaných schém a programových knižníc môže človek zaobstarať a vyhotoviť sám. Tieto zariadenia sú napojené na mikrokontrolér Uno R3+WIFI, aby bola zabezpečená kompatibilita programových knižníc a zároveň bola poskytnutá možnosť odosielat' dátu zo zariadenia priamo na Cloud bez sietového pripojenia, ale bezdrôtovo. Mechatronické

zariadenia boli autorkou prispôsobené tak, aby vedeli bezdrôtovo komunikovať so vzdialou výpočtovou jednotkou umiestnenou na cloude od poskytovateľa služieb Google Cloud Platform. Špecifikácia zariadení a spôsob ich úpravy je uvedený v nasledovných podkapitolách.

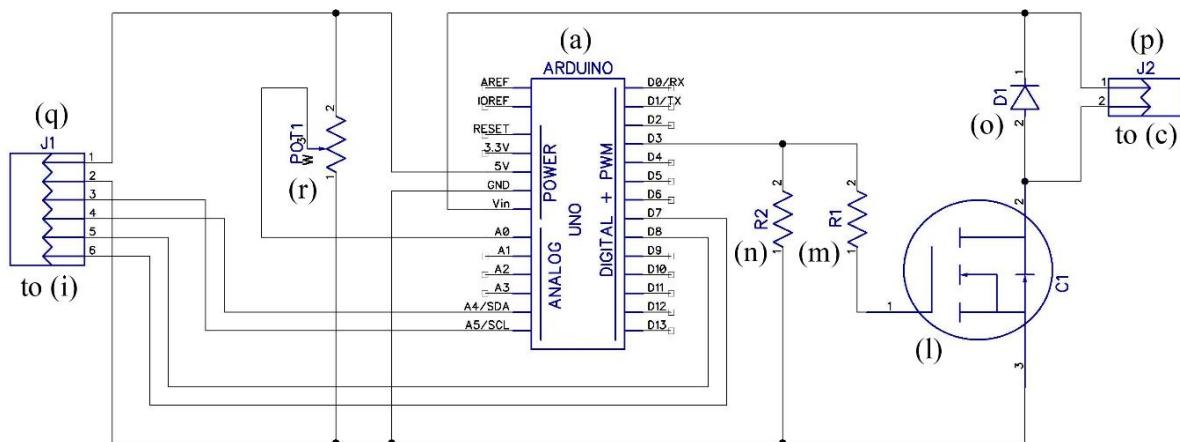
6.1.1 Vlastnosti a špecifikácia zariadení

Základný dizajn **FloatShield** sa skladá z prieľadnej vertikálnej trubice obsahujúcej loptičku, ktorá sa vznáša vo vnútri v dôsledku prúdenia vzduchu generovaného ventilátorom pripojeným ku konštrukčnej doske plošných spojov.



Obr. 34 Popis zariadenia FloatShield [37]

Trubica a senzor vzdialenosť sú prichytené pomocou 3D vytlačených častí. Cieľom je riadiť polohu lopty v trubici zmeranú senzorom vzdialenosť, ktorý vytvára jednoduchú spätnoväzbovú slučku s jednoduchým vstupom (SISO). Užívateľ môže tiež regulovať výkon ventilátora manuálne - pomocou potenciometra, ktorého výstup je potom upravený na signál modulovaný šírkou impulzu (PWM).



Obr. 35 Obvod zapojenia zariadenia FloatShield a Arduino vývojovej dosky [37]

Obr. 35 popisuje zapojenie obvodu jednotlivých elektronických komponentov na samotnú vývojovú dosku Arduino (v našom prípade UNO R3+WIFI). Ventilátor a snímač TOF sú reprezentované iba ich konektormi, pričom predpokladáme, že 12 V externý výkon je privádzaný cez VIN pin Arduino (a). Ventilátor je napájaný N-kanálovým MOSFETom C1 (l) a je poháňaný pinom mikrokontroléra s podporou D3 PWM cez odpor odporu R1 (m) s prúdom $1\text{ k}\Omega$. Pohyblivé stavy sú riešené $10\text{ k}\Omega$ odporom R2 (n), zatiaľ čo dióda D1 (o) zaistzuje ochranu proti spätej elektromotorickej sile (EMF). Konektor (p) nakoniec vedie k terminálom ventilátora. Pretože ventilátor vyžaduje 12 V a viac prúdu, ako dokáže Arduino s napájaním USB zvládnuť, na prevádzku zariadenia je potrebný samostatný zdroj napájania z adaptéra. Senzor TOF je integrovaný do vhodnej otvorenej breakout dosky, takže môže byť bez námahy pripojený (q) k zberniči I2C Arduino (SDA, SCL). Nakoniec je bežec potenciometra POT1 (r) pripojený k pinu dosky A0, ktorá podporuje ADC (analógovo-digitálny prevodník).

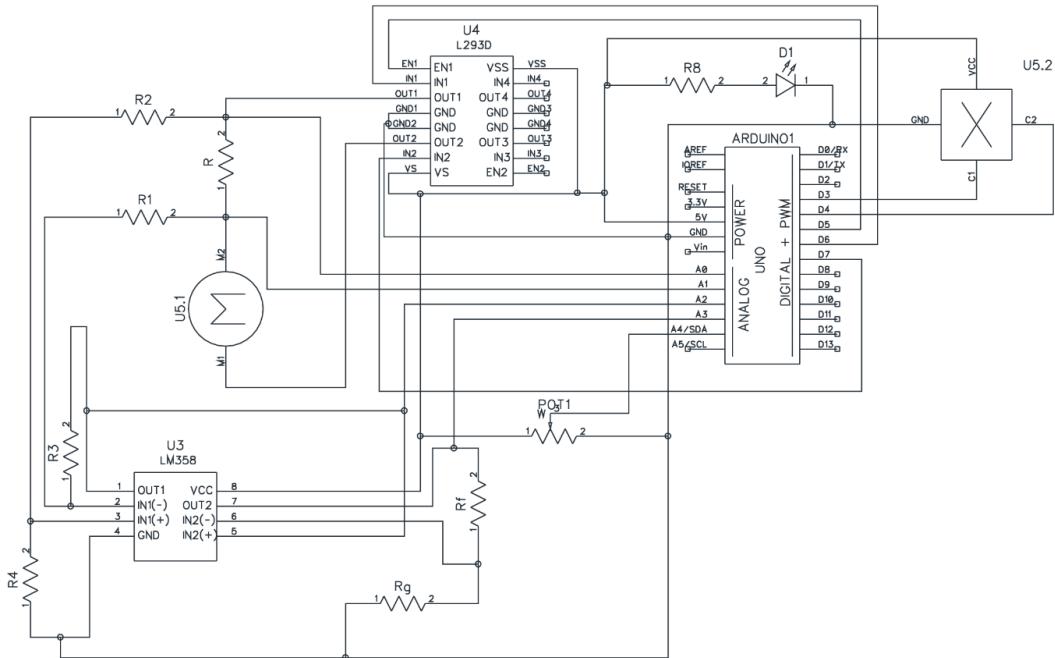
MotoShield implementuje motor vybavený redukciou prevodového stupňa ako ovládač a spätnú väzbu enkodéra. MotoShield umožňuje uskutočňovať množstvo experimentov v oblasti riadenia, od modelovania prvotných princípov a identifikácie systému ako šedej

skrinky, riadenia rýchlosťi a riadenia polohy. Rýchlosť otáčok motora môže používateľ regulovať pomocou potenciometra.



Obr. 36 Ukážka zariadenia MotoShield [37]

Hlavnou časťou MotoShieldu je jednosmerný motor s kefkou 6 V vybavený halovým snímačom polohy. Motorová jednotka má 6 výstupov, z ktorých 4 patria k snímačom. L33D H-mostík integrovaný obvod sa používa ako ovládač motora. L293D je veľmi praktický a ľahko použiteľný integrovaný obvod obsahujúci štvor kanálový H-mostík, ktorý mu umožňuje ovládať dva jednosmerné motory, jeden krokový motor alebo štyri ďalšie záťažové jednotky, ako sú solenoidy alebo relé. MotoShield tiež obsahuje operačný zosilňovač LM358, ktorý má dve úlohy. Najprv odpočíta dve hodnoty napäťia, a tým vypočíta úbytok napäťia rezistorom použitým na meranie prúdu. Po druhé, zosilňuje odčítané napätie Arduina [37].



Obr. 37 Obvod zapojenia zariadenia MotoShield a Arduino vývojovej dosky [37]

6.1.2 Ovládanie zariadenia

Ovládanie zariadenia je možné naprogramovať pomocou C/C++ v programe Arduino IDE. Programovanie riadenia na ATMega328 je potrebné dať do samostatného projektu a programovanie riadenia pre ESP8266 si vyžaduje vlastný projekt, ktorý sa na čip nahrá. Rozdielnosť je spôsobená odlišnou funkcionálitou jednotlivých čipov, ktoré následne medzi sebou komunikujú. Zatiaľ čo ATMega328 sa stará o priame ovládanie zariadenia, tak ESP8266 sa stará o komunikáciu zariadenia s online clouдовou výpočtovou jednotkou.

AutomationShield knižnica

Základné aplikačné programovacie rozhranie (API) slúžiace zariadeniu je napísané v C/C++ a je integrované do voľne dostupnej Arduino knižnice AutomationShield. Táto knižnica obsahuje hardvérové ovládače a ukážkové cvičenia pre technické vzdelávanie riadiacich systémov. Všetky funkcie spojené s FloatShield a MotoShield sú zahrnuté v hlavičke FloatShield.h a MotoShield.h. Tieto hlavičky obsahujú triedy FloatClass a MotoClass, ktoré sú štandardne konštruované ako objekty FloatShield a MotoShield. Funkcie špecifické pre tieto štíty väčšinou vykonávajú periférnu komunikáciu vstup/výstup.

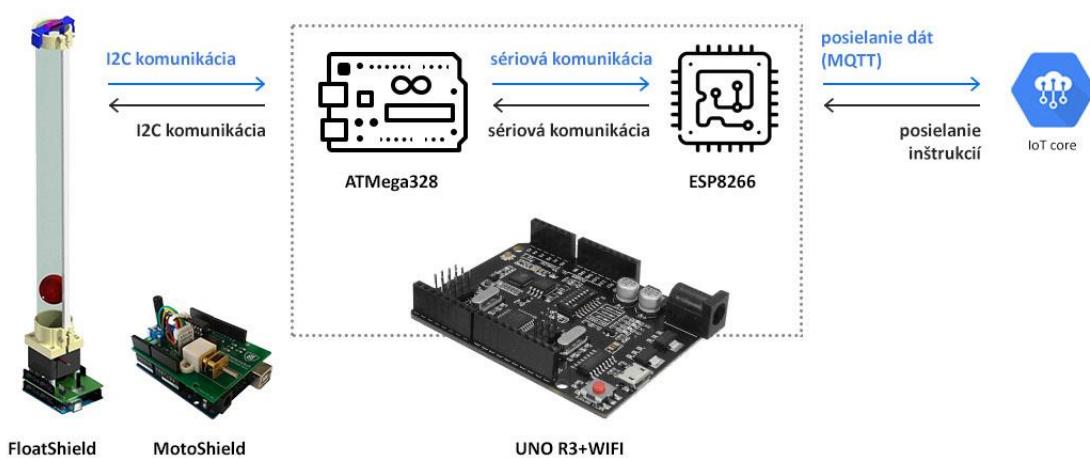
Google Cloud IoT JWT knižnica

Oficiálna knižnica pre Arduino pre komunikáciu zariadenia so službou Google IoT Core s podporou čipov Genuino MKR1000, WiFi101, Espressif ESP32 a Espressif ESP8266.

Táto knižnica sa skladá z dvoch hlavných komponentov: knižnice na vytváranie JWT (JSON web token) pre autentifikovanie zariadenia so službou Google IoT Core, pričom je potrebné vytvoriť EC privátny / verejný kľúč pre zariadenie. Druhá časť knižnice spočíva v komunikačnom protokole MQTT, ktorú je potrebné vopred inštalovať. Jej názov je IwMQTT. Obidve tieto knižnice sú dostupné v manažéri knižníc v rámci prostredia Arduino IDE. Táto knižnica obsahuje vo svojich ukážkových kódoch zdrojový kód s názvom „ciotc_config.h“, ktorý je potrebné vyplniť podľa vlastných údajov. Bližšie sa jej budeme venovať v kapitole 6.3.2.

6.1.3 Komunikácia zariadenia s clandom

Obr. 38 opisuje komunikáciu medzi mechatronickým zariadením a clandom. Samotný shield je zariadenie, ktoré sa priamo napája na vývojovú dosku tak, že pokrýva všetky vstupné PINy. Shieldy z rodiny AutomationShield komunikujú so samotnou doskou pomocou I2C protokolu. Na doske UNO R3+Wifi sa nachádzajú 2 čipy a to ATMega328 a ESP8266. Shieldy komunikujú s čipom ATMega328, nakoľko tento čip má vstupné PINy s ním kompatibilné. PINy ESP8266 majú iné umiestnenie a charakter. PINy ATMega328 majú charakter samice, takže je možné naň Shield umiestniť, zatiaľ čo PINy čipu ESP8266 majú charakter samca. Tieto dva čipy na doske nie je potrebné prepájať dodatočným zapojením, nakoľko pri prepnutí spínača na doske do správnej polohy, sú tieto čipy schopné komunikovať medzi sebou pomocou sériovej komunikácie.



Obr. 38 Zariadenie FloatShield / MotoShield a jeho komunikácia s clandom

Sériová komunikácia medzi čipmi závisí od vopred stanovených správ.

Tabuľka 4 Komunikácia medzi ESP8266 a ATMega328 pre FloatShield a MotoShield

Zariadenie	Odosielateľ Prijímateľ	Príkaz	Popis
FloatShield	ATMega328 ESP8266	<18.2, 77.3>	aktuálna výška loptičky [mm], otáčky ventilátora [%]
MotoShield	ATMega328 ESP8266	<18.2, 77.3, 80.7, 30.7>	úbytok napäťia rezistorom [V], aktuálny odber motora [mA], rýchlosť otáčok [%], nastavená rýchlosť motora PID regulátorom

Mechatronické zariadenia majú na svojich oficiálnych stránkach definované základné príkazy pre ich monitorovanie a riadenie, avšak len na úrovni samotného zariadenia. To znamená, že zariadenia dátá nikam neodosielajú a je možné ich ovládať iba pomocou potenciometra, alebo mu vopred definovať jeho optimálne správanie. Preto sme navrhli a implementovali vlastný spôsob komunikácie založenom na sériovej komunikácii dvoch čipov, pričom jeden sa stará o komunikáciu s mechatronickým zariadením a druhý sa stará o komunikáciu s cloudom.

6.2 Inteligentné vyhodnocovanie stavu zariadenia

Podkapitola „Inteligentné vyhodnocovanie stavu zariadenia“ opisuje charakter nameraných dát zo sledovaných zariadení, návrh a implementáciu vlastných segmentačných algoritmov, ktoré tieto dátá využívajú a spôsob, akým tieto autorkou navrhnuté a spracované inteligentné algoritmy vyhodnocujú stav sledovaného zariadenia.

Inteligentná diagnostika porúch zariadení využíva niektoré schopnosti umelých neurónových sietí, ako sú extrakcia vlastností, segmentácia a klasifikácia. Na základe údajov senzora je možné extrahovať vlastnosti a vzorce detegovaného signálu. Extrahované znaky opisujú štruktúru zachytených signálov a môžu sa použiť na segmentáciu signálu naprieč jednotlivými časovými bodmi a na konci tieto kroky klasifikujú do vopred definovaných kategórií. Tieto 3 kroky na úspešnú detekciu a klasifikáciu poruchy zariadenia sú riešiteľné prostredníctvom hlbokého učenia, v ktorom kombinácia vytvára inteligentné riešenie diagnostiky zariadenia.

Najrelevantnejšie informácie na rozpoznanie poruchy zariadenia sú časovo-frekvenčné vlastnosti extrahované zo zaznamenaných signálov zo snímačov zariadenia. Vzhľadom na

celý rad rôznych typov snímačov a sledovaných signálov, je možné povedať, že všetky majú jednu dimenziu rovnakú, a to čas. V zmysle hlbokých neurónových sietí existuje niekoľko typov blokov, medzi ktorými sú takzvané konvolučné neurónové siete (CNN) a rekurentné neurónové siete (RNN) vhodné na riešenie problému časových závislostí. RNN sa zvyčajne používajú v problémoch časových závislostí, pretože archivujú informácie o niekoľko krokov späť vo svojich jednotkách. Široko používanou formou RNN je dlhá krátkodobá pamäť (LSTM), ktorá prekonáva problém dlhodobých závislostí. Veľkou nevýhodou RNN je ich nízka výpočtová rýchlosť a problematické použitie paralelizmu. Hlavnou schopnosťou CNN je extrahovať vlastnosti zo vstupných údajov, či už sú to 2D dátá, ako sú obrázky, alebo 1D dátá, ako sú signály a jednosmerne orientované údaje. Hlavnou myšlienkou CNN je konvolučný proces, pri ktorom sa kombinujú filtre a vstupné údaje a konečný produkt predstavuje aktivačnú mapu vstupu. Popri CNN sa pri extrakcii prvkov poruchy zariadenia používa aj metóda zvaná auto-enkóder. Auto-enkóder slúži ako extraktor vlastností a rekonštrukciou dát na základe extrahovaných vlastností. Základným princípom auto-enkódera je zobrať uzly vstupnej vrstvy a spojiť ich do nižšieho počtu uzlov v skrytej vrstve, ktoré predstavujú extrahované prvky. Auto-enkóder rekonštruuje vstupné uzly na výstupných uzloch na základe vlastností získaných v skrytej vrstve. Auto-enkóder sa zvyčajne používa ako dátový kompresor na ukladanie veľkorozmerných údajov a ich reprezentáciu pomocou komprimovaných verzií. Tieto komprimované verzie predstavujú vlastnosti vstupných údajov. Po získaní vlastností je možné kategorizovať naučené vlastnosti a klasifikovať vždy, keď je signál chybný a ktorú poruchu predstavuje.

Namerané dátá zo zariadení, ako aj natrénované modely a zdrojové kódy je možné nájsť v prílohách dizertačnej práce.

6.2.1 Dáta definujúce zariadenia

Sledované zariadenia je možné monitorovať na základe dát zo senzorov. Preto, jednotlivé dátá meriame a merané hodnoty ukladáme. Pre algoritmy hlbokého učenia v prípade učenia s učiteľom je potrebné tieto dátá označiť a zaradiť do kategórií definujúcich stav zariadenia. Dátá označujeme a následne štandardizujeme aby sa predišlo zameriavaniu umelých neurónových sietí na okrajové hodnoty, ktoré sú výrazné svojou veľkosťou, avšak nemusia mať rovnako veľkú aj dôležitosť reprezentácie dát.

Nakoľko máme nie-len jednu sledovanú hodnotu, ale zameriavame sa na komplex sledovaných parametrov meniacich sa v čase, ktoré samotný mechatronický systém opisujú, mení sa nám charakter dát z 1D orientovaných na 2D. Samotnú prácu s dátami sme rozdelili do nasledovných krokov:

1. Získanie dát z reálneho prostredia.
2. Vzorkovanie dát – rozdelenie dát podľa kategórií na časové okná.
3. Rozdelenie dát na trénovacie a validačné.
4. Štandardizácia dát.

Dáta sme získali namerali experimentovaním počas reálneho chodu zariadení v rôznych podmienkach vytvárajúcich chyby na zariadení. Definovali sme niekoľko typov chýb – odchýlok od normálneho režimu, na ktorých detekciu sa zameriame v návrhu inteligentného systému. Sledované hodnoty pre zariadenia boli nasledovné:

- FloatShield – pozícia loptičky [mm], výkon ventilátora [%].
- MotoShield - úbytok napäťa rezistorom [V], aktuálny odber prúdu motora [mA], rýchlosť otáčok [%], nastavená rýchlosť motora PID regulátorom [%].

Pre FloatShield sme zaznamenali viac ako 1 milión časových bodov zachytávajúcich 6 druhov stavov: 1 bezchybný stav a 5 chybových stavov. Stavy, ich definície, označenie a percentuálny podiel v súbore experimentálnych dát sú opísané v Tabuľka 5. Pre MotoShield sme zaznamenali viac ako 2,5 milióna časových bodov zachytávajúcich 4 stavy: 1 bezchybný stav a 3 chybové stavy, pričom tieto chyby sa mohli vyskytovať v rôznych intenzitách a na rôznych častiach zariadenia. Definície a označenie chýb ako aj percentuálny podiel v celkovom experimentálnom datasete sú zobrazené v Tabuľka 6.

Tabuľka 5 Kompozícia datasetu pre diagnostiku zariadenia FloatShield

Trieda	Popis	Podiel [%]
0	Bez chyby	27
1	Stredne ľažký predmet padol do tuby	45
2	Ľažký predmet padol do tuby	3
3	Ľahký predmet padol do tuby	18
4	Zaseknutá loptička	3
5	Loptička uvoľnená po zaseknutí	4

Tabuľka 6 Kompozícia datasetu pre diagnostiku zariadenia MotoShield

Trieda	Popis	Podiel [%]
0	Bez chyby	49
1	Zaseknuté teleso v ozubených kolieskach	25
2	Pôsobenie magnetu	25
3	Blokovaný motor	1

Ako je vidieť v predchádzajúcich tabuľkách, datasety sú značne nevyvážené. Za bežných okolností je zariadenie v bezchybnom stave väčšinu času, preto spadá väčšinu času do práve jednej kategórie. Pre simulovanie rôznorodosti sú aj samotné datasety rôznorodé a nevyvážené. Riešenie problému trénovania algoritmov v prípade nevyváženého objemu záznamov pre rôzne stavy vzbudzuje v posledných rokoch záujem vzhľadom na exponenciálny nárast objemu získavaných informácií ako dôsledok digitalizácie v priemysle. S nárastom objemu dát sa pritom prehlbuje aj ich nevyváženosť.

Jedna vzorka zo získaného datasetu dosahuje rozmery (t, n) , kde hodnota t predstavuje počet sledovaných bodov času v rámci časového okna a hodnota n počet sledovaných parametrov. Ako dĺžka časového okna bola zvolená hodnota 256 kontinuálnych bodov času. Tieto časové okná sú vyberané zo získaných záznamov správania sa systému postupne pomocou pohybujúceho sa okna s krokováním 128 bodov času. FloatShield aj MotoShield majú nastavené krokovanie na 25ms, čo v prípade 256 bodov času predstavuje záznam o dĺžke približne 6,4 sekundy. Časové okno pre FloatShield teda naberá rozmery (256, 2) a pre MotoShield (256, 4).

Potom, čo sú dáta prerozdelené na jednotlivé vzorky fixnej dĺžky, je potrebné získané údaje rozdeliť na testovaciu a validačnú zložku. Údaje boli rozdelené do trénovacieho a validačného setu v pomere 90% záznamov v trénovacom sete a 10% vo validačnom sete proporcionálne pre každú triedu, aby sa zabezpečilo, že každá trieda má zástupcov tak v trénovacom sete, ako aj vo validačnom sete.

Posledným krokom pri spracovaní dát pred trénovaním NN je normalizácia dát z trénovacieho aj validačného datasetu. Normalizácia dát je potrebná v prípade rozličného rozsahu parametrov, aby sa zabránilo favorizovaniu údajov s vyššími hodnotami, aj keď menšou výpovednou hodnotou. Pre tieto účely bola vybraná normalizácia pomocou Z-skóre, ktoré využíva na úpravu hodnôt strednú hodnotu a štandardnú deviáciu. Stredná hodnota a štandardná deviácia sa vypočítajú iba pre trénovací dataset. Následne sa princíp

normalizácie aplikuje na trénovací dataset, validačný dataset a každé ďalšie namerané veličiny pripravené na predikciu práve s týmito hodnotami. Z-skóre je definované ako:

$$z_i = \frac{x_i - \bar{x}}{s} \quad (10)$$

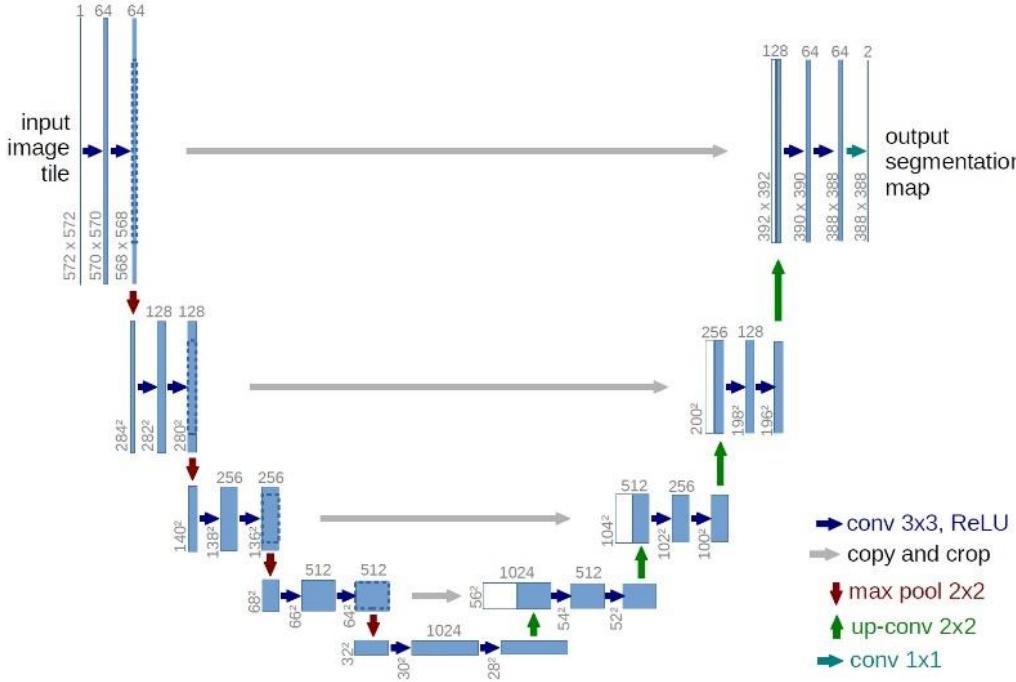
kde z_i označuje štandardizovanú hodnotu, x_i pôvodnú hodnotu, \bar{x} je stredná hodnota a s je štandardná deviácia.

6.2.2 Architektúra umelej neurónovej siete

V podkapitole „Architektúra umelej neurónovej siete“ autorka dizertačnej práce navrhla a spracovala vlastný segmentačný algoritmus modifikáciou už existujúcich inteligentných algoritmov.

V kapitole 3 Inteligentná diagnostika v priemysle sme analyzovali niekoľko výskumných prác v tomto smere za obdobie posledných piatich rokov. Výskum v oblasti zdravia mechatronických systémov sa zväčša zameriava na jednu sledovanú veličinu a sledovaný časový segment je vždy vyhodnocovaný ako celok, či menší blok. Je možné sledovať použitie ako jednoduchých NN, tak aj CNN a RNN. Vzhľadom na skutočnosť, že RNN je pomalá na trénovanie a nemusí v konečnom dôsledku dosahovať priaznivé výsledky bez kombinácie s CNN, rozhodli sme sa RNN do výsledného algoritmu nezaradiť. Nakol'ko cieľom nášho algoritmu je diagnostikovanie zdravia zariadenia v každom bode sledovaného času pre včasné odhalenie chyby, využili sme princíp segmentácie. Segmentácia je založená na získaných vlastnostiach, ktoré nám naše dátá opisujú a preto sme sa rozhodli zvoliť konvolučné bloky, ako podstatu diagnostického algoritmu, ktoré sú na hľadanie vlastností v nelineárne orientovaných dátach ideálne.

V prípade detekcie a klasifikácie porúch sa obvykle na extrahovanie vlastností vstupných údajov a klasifikáciu výstupu používajú zložené alebo riedke auto-enkodéry. V zmysle segmentácie sa používa auto-enkodér s maskovanými údajmi na klasifikáciu každej jednotky vstupného segmentu a nielen segmentu ako celku. Príkladom segmentovania auto-enkóderom je architektúra hlbokej neurónovej siete s U-tvarom nazývaná U-Net. U-Net je neurónová sieť navrhnutá pre sémantickú segmentáciu. Pôvodne určená na segmentáciu lekárskych obrazov [39], ale preukázala sa tiež ako účinná na spracovanie signálu [40].



Obr. 39 U-Net architektúra so vstupom 572×572 [39]

Architektúra U-net je plne konvolučná neurónová siet' pozostávajúca z blokov konvolúcii, max poolingu, dekonvolúcii a spájania. Pre kombinovanú segmentáciu signálu bola použitá a autorkou dizertačnej práce adaptovaná architektúra U-Net. Navrhnutú modifikáciu architektúry, ktorá je jedným z pôvodných prínosov práce opisujeme v nasledujúcich odsekoch.

Architektúra U-Net sme zmenili pridaním dávkovej normalizácie po každej konvolúcii a pridávania dropoutu (miery odstránenia prepojení uzlov medzi vrstvami) v konvolučnom bloku po prvej konvolúcii, ako je to vidieť na Obr. 40. Samotný dropout sa mení každou hlbkovou vrstvou označujúcu hlbku samotného modelu siete U-Net. Čím hlbšie je enkodér U-Net, tým sa dropout postupne zvyšuje podľa vopred definovanej zmeny dropoutu. V dekódovacej časti dropout neustále klesá na základe hlbky siete. Podstatným nastavením, ktoré zaručuje variabilitu riešenia naprieč systémom s rôznym počtom sledovaných hodnôt je veľkosť pooling okna. Nakoľko v konečnom dôsledku chceme dostať výslednú predikciu ako maticu o veľkosti $(t, 1, c)$, kde t označuje počet časových bodov, 1 je na miesto počtu sledovaných hodnôt, pretože nás nezaujíma odpredikovaný stav pre každú premennú, ale pre systém ako taký v konkrétnom časovom bode, a napokon c , ktoré označuje počet tried / stavov, ktoré môže zariadenie nadobúdať. V každom bode tohto objektu dostávame

pravdepodobnosť konkrétneho stavu v konkrétnom časovom bode. Aby sme takýto výstup z neurónovej siete dostali, tak sme U-Net architektúru modifikovali.

Nech máme ako vstup do neurónovej siete, ktorý chceme segmentovať, maticu M . Veľkosť matice M nech je daná hodnotou M_s ako:

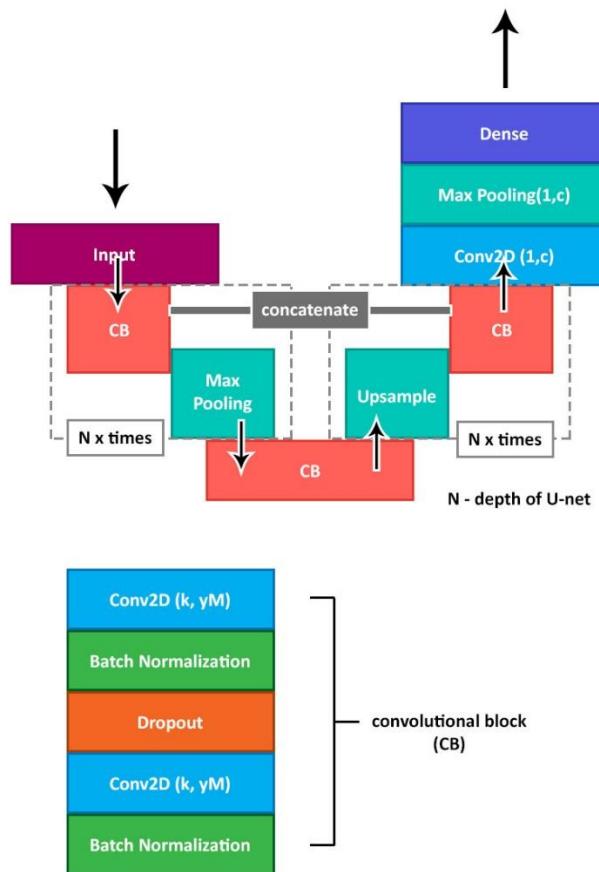
$$M_s = (x_M, y_M) \quad (11)$$

kde hodnota x_M predstavuje šírku matice M a y_M predstavuje výšku matice M . Nech máme k vstupnej matici M vstupnú masku vektor V_M . Tento vektor predstavuje segmentačný vektor po osi x a tak nadobúda veľkosť šírky matice M , x_M . Tento vektor môže nadobúdať hodnoty $< 0, c$), kde c označuje počet segmentačných tried. Maticu M následne rozšírimo o ďalšiu dimenziu, čím ju zmeníme na objekt \bar{M} , a jej veľkosť \bar{M}_s bude nadobúdať rozmery $(x_M, y_M, 1)$. Vektor V_M upravíme pomocou metódy „one hot encoding“ na \bar{V}_M tak, aby nadobúdal iba hodnoty 0 alebo 1 a rozšírimo ho o ďalšie dve dimenzie tak, aby nadobúdal rozmery $(x_M, 1, c)$, čím ho zmeníme na objekt s rovnakým počtom osí, ako objekt \bar{M} . Úlohou neurónovej siete je určiť masku objektu \bar{M} tak, aby sa zhodovala s jeho skutočnou maskou \bar{V}_M a nadobúdala rozmery $(x_M, 1, c)$. Klasická U-Net architektúra avšak predpovedá masku o veľkosti $(x_M, y_M, 1)$. V rámci architektúry a jej nastavení sme vykonali nasledujúce zmeny:

- Nastavenie veľkosti filtra pri konvolúcii na (k, y_M) , kde k označuje číslo v rozmedzí $\langle 1, x_M \rangle$.
- Nastavenie veľkosti okna poolingu na $(c, 1)$.
- Zmenu poslednej konvolúcie tak, že veľkosť filtra nebude $(1, 1)$ ale otočená veľkosť poolingu, t.j. $(1, c)$ a aktivačná funkcia bude ReLu.
- Za poslednou konvolúciou bude nasledovať max pooling s otočenou veľkosťou filtra, t.j. $(1, c)$.
- Za max poolingom nasleduje výstupná plne prepojená vrstva s počtom uzlov c .

Vo výsledku dostávame segmentačný algoritmus, ktorý slúži ako segmentátor po x-ovej osi a nazývame ho **xAxis U-Net**. Jeho architektúra je znázornená na Obr. 40 **xAxis U-Net**. **xAxis U-Net** je možné využiť aj na segmentovanie po y-ovej osi a to tak, že bude rozdielny oproti segmentačnom algoritme **xAxis U-net** v nasledovných bodoch:

- \bar{V}_M bude nadobúdať rozmery $(1, y_M, c)$.
- Veľkosť konvolučného filtra bude (x_M, k) .
- Veľkosť okna poolingu bude $(1, c)$ a tým pádom, veľkosť otočeného okna poolingu bude $(c, 1)$.



Obr. 40 xAxiS U-Net

Na základe výsledkov algoritmu Ax search bola vybraná zostava modelu. Ax search je vyhľadávací algoritmus pre hyperparametre modelu. Hľadaním bolo vybraných niekoľko hyperparametrov, ako napríklad: použitie dávkovej normalizácie, hĺbka xAxiS U-Net siete, veľkosť konvolučného jadra k , počet filtrov v prvej hĺkovej vrstve, pretože ich počet sa zvyšuje každou hĺkovou vrstvou násobenou poradovým číslom, rýchlosť učenia, dropout a zmena dropoutu do hĺbky. Ako optimalizačný algoritmus sme vybrali ADAM. Výber vhodných hyperparametrov a trénovanie navrhnutej architektúre je rozvinuté v kapitole 7.

Vzhľadom na vysoko nevyvážený dataset, je potrebné zvoliť adekvátnu chybovú funkciu, ako aj evaluačnú metriku. Keďže problém spočíva v definícii viacerých tried, typickým prístupom by bolo použitie kategorickej krízovej entropie ako chybovej funkcie, ale keďže je dataset veľmi nevyvážený, namiesto toho sme použili funkciu s názvom „focal loss“. Focal loss navrhnutá [41] sa zaoberá nerovnováhou pridaním váhového faktora α_t a parametra zaostrenia γ_t , ktorý sa končí definíciou ohniskovej straty ako:

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (12)$$

kde $FL(p_t)$ označuje ohniskovú stratu podľa modelom odhadovanej pravdepodobnosti pre triedu s označením $t, p_t \in [0,1]$. Bola použitá chybová funkcia focal loss s odporúčanými hodnotami $\alpha_t = 0,25$ a $\gamma_t=2,0$.

Ako už bolo spomenuté, dataset je vysoko nevyvážený, a preto sa metrika hodnotenia opisujúca účinnosť poskytovaného riešenia musí zvoliť odlišne a to s dôrazom na skutočnú povahu údajov. V štúdiách [42] a [43] sa Matthewsov korelačný koeficient (MCC) zvolil ako vhodná metrika oproti AUC, Cohenov Kappa a F1 skóre pre vyhodnotenie úspešnosti modelu získanú z nevyváženého datasetu. Matthewsov korelačný koeficient sa rovná:

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp+fp)(tp+fn)(tn+fp)(tn+fn)}} \quad (13)$$

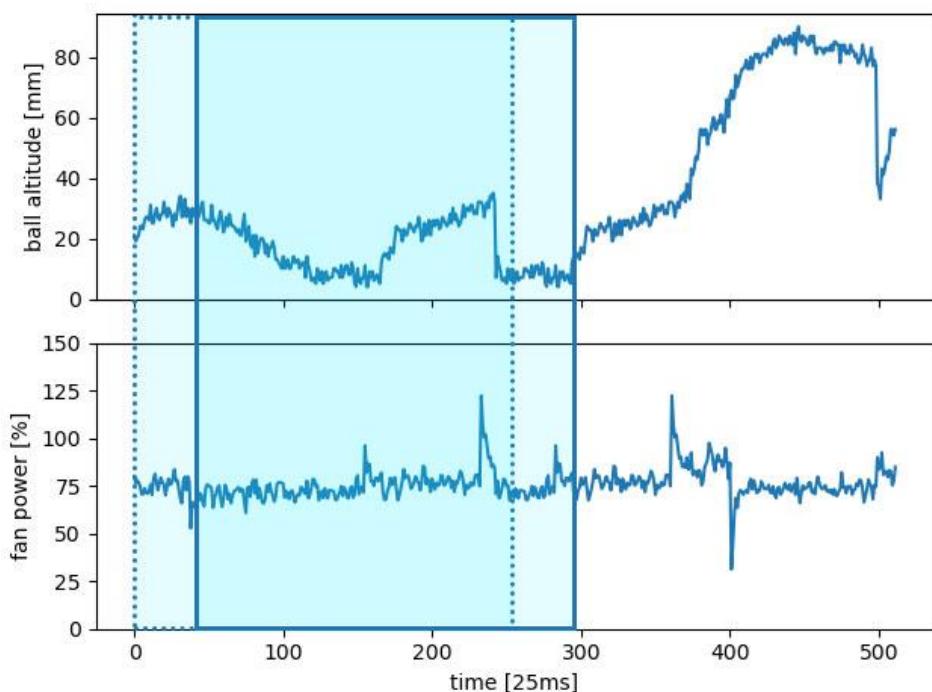
kde tn označuje skutočne nesprávne, tp sú skutočné správne, fn sú nesprávne nesprávne a fp sú nesprávne správne. Triedy signálov sú reprezentované ako polia dĺžky rovnajúce sa celkovému počtu tried pozostávajúcich iba z núl a jednotiek. Tam, kde sa vyskytuje číslo 1, to označuje index triedy, do ktorej signál prislúcha.

6.2.3 Princíp vyhodnocovania stavu zariadenia navrhnutým riešením inteligentnej diagnostiky

Zaznamenané hodnoty zo zariadenia budú postupne predikované tak, že obsiahnu veľkosť časového okna 256 záznamov, tak ako bol model trénovaný, čo predstavuje približne 6,4 sekundy. Výsledok predikcie modelu je objekt o veľkosti $(t, 1, c)$, kde t označuje veľkosť časového okna vzhľadom na počet záznamov a c označuje počet stavov. V každom bode

tejto matice je uchovaná hodnota v rozmedzí 0 až 1. Táto hodnota označuje pravdepodobnosť konkrétneho stavu zariadenia v danom časovom bode. Tento objekt pravdepodobností je následne konvertovaný na vektor hodnôt o dĺžke t , kde v každom bode času sa nachádza hodnota označujúca index stavu zariadenia, ktorému bol daný časový bod určený. Toto určenie stavu sa vyhodnotí ako index najvyššej hodnoty v čase bodu t .

Nakoľko vyhodnocovanie stavu zariadenia každých 6,4 sekundy nemusí byť dostatočne efektívne, tak je stav zariadenia vyhodnocovaný od prvých 6,4 sekundy od zaznamenávania každú sekundu, čo predstavuje vždy posun o 40 hodnôt. Takúto metódu predikcie je možné označiť ako predikciu klízavým oknom. Týmto spôsobom sa zaistí aktuálnosť údajov a prípadná korekcia nesprávne predikovaných hodnôt priemerovaním predikcií pre konkrétny časový bod.



Obr. 41 Predikcia klízavým oknom

6.3 Google Cloud Platform (GCP)

Cieľom tejto podkapitoly je návrh a implementácia vlastného riešenia komunikácie mechatronického zariadenia s cloudom a cloudovej realizácie intelligentnej detekcie stavu. Riešenie je navrhnuté ako bez-serverové.

Google Cloud Platform je rozhranie, ktoré nám poskytuje široko-spektrálne cloudové služby. Na to, aby sme dokázali vytvoriť bez-serverový projekt, ktorý by nám poskytoval komunikáciu mechatronického zariadenia a diagnostického systému založeného na umelej inteligencii budeme potrebovať niekoľko služieb, ktoré toto rozhranie ponúka. Budeme potrebovať registrovať zariadenie, ako spárované v rámci IoT štruktúry pomocou IoT Core. Toto zariadenie bude komunikovať s cloudovými službami pomocou Pub/Sub komunikačného toku a preto bude potrebné aktivovať službu rovnakého názvu. Naše zaznamenané dátá si budeme ukladať do vhodnej databázy, akou je BigQuery. Na to, aby sme do tejto databázy uložili prichádzajúce údaje a zároveň aj predikcie umelej inteligencie definujúce stav zariadenia, využijeme Cloud Functions. Bez potreby fyzického či virtuálneho servera, budeme môcť tieto funkcie vždy spúšťať pri určitej udalosti. Aby sme mohli využívať nami natrénovaný model pri online predikciách, budeme potrebovať si ho spolu so zdrojovým kódom potrebným na predikciu uložiť v cloude a to v časti storage ako takzvaný bucket. Následne, keď máme zdrojové kódy aj s natrénovaným modelom umiestnené na cloude, môžeme k nim vytvoriť takzvanú rutinu, kód, ktorý sa spúšťa na požiadanie. V tomto prípade ide o rutinu z rodiny AI platform, v rámci ktorej si vytvoríme obsluhovateľný model.

Na to, aby sme mohli nastaviť GCP služby, budeme potrebovať:

- Prístup na stránku Google Cloud Platform
- OpenSSL
- Python environment

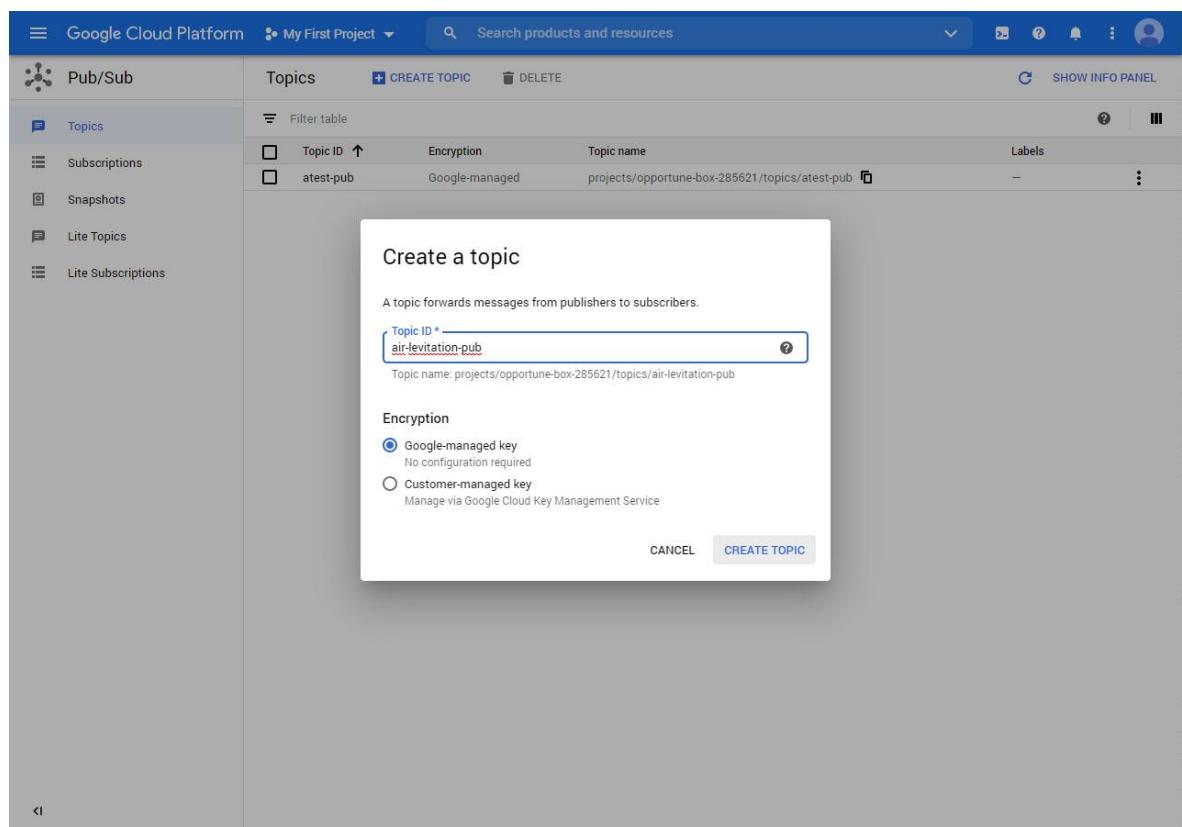
Ako užívateľsky príjemnejšia možnosť je využitie stránky Google Cloud Platform, no je možnosť vytvárania jednotlivých služieb a ich inštancií aj pomocou príkazového riadku buď priamo na stránke v administrátorskom rozhraní zapnutím GCP Shell, alebo stiahnutím si GCP SDK. V prípade ukážok budeme ďalej v dizertačnej práci uvádzať prácu s GCP cez grafické rozhranie online stránok.

Po vytvorení konta v GCP je potrebné vytvoriť aj náš prvý projekt. Názov tohto projektu je nemenný a tak je potrebné si dobre zvážiť jeho názov. Ako ukážku si vytvoríme jednotlivé komponenty a prepojenia pre FloatShield. Potrebné kódy pre MotoShield je možné nájsť v prílohe dizertačnej práce.

6.3.1 GCP Pub-Sub

Princíp Pub/Sub komunikácie spočíva v dvoch hlavných konceptoch: publikácie na určitú tému – topic, a potom odoberanie určitej témy – subscription. To znamená, že keď zariadenie chce odosielat' údaje, je v úlohe publikovateľa a údaje odosielá na určitú tému. Ďalšie zariadenia, ktoré odoberajú danú tému tieto údaje prečítajú.

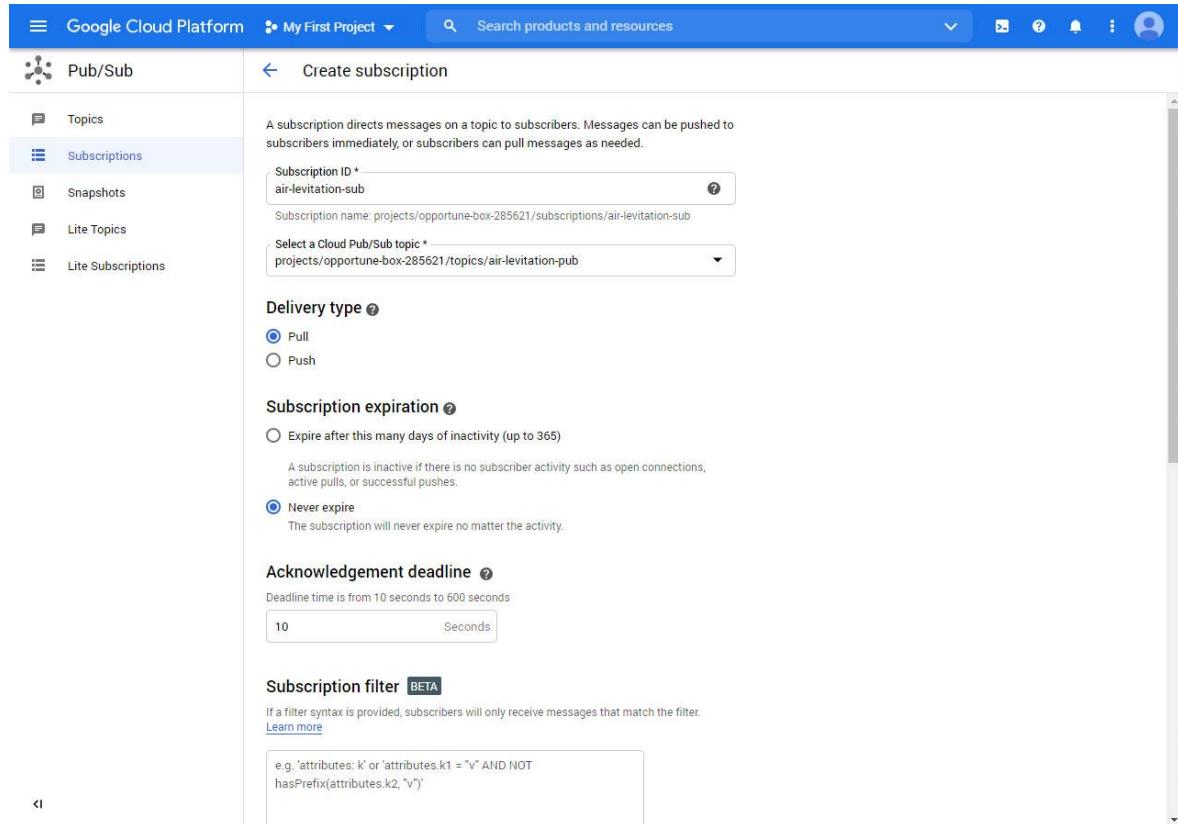
V koncepte navrhnutého systému je zariadenie spárované s GCP na základe prijímania a odosielania dát práve na témach správ. Preto je potrebné si vytvoríť tému publikovania a tému odoberania v rámci nášho projektu. V GCP pôjdeme do sekcie BigData - Pub/Sub a tam si zvolíme záložku „Topics“. Ako Topic ID si zvolíme „air-levitation-pub“ a dáme „create topic“.



Obr. 42 Vytvorenie témy pre publikovanie

Aby sme mohli dáta nie len posielat', ale aj prijímať, vytvoríme aj odoberanie témy v záložke „Subscriptions“ a dáme „create subscription“. Subscription ID bude „air-levitation-sub“, Cloud Pub/Sub si zvolíme nás práve vytvorený topic. Expiráciu si nastavíme

na „never expire“ čiže bez expirácie a typ prijatia ako „Pull“. Ostatné polia necháme predvyplnené.



Obr. 43 Vytvorenie témy na odoberanie

6.3.2 GCP IoT Core

Aby sme vedeli zariadenie autorizovať v prípade, že sa bude chcieť pripojiť, je potrebné najprv cez OpenSSL na osobnom počítači vygenerovať EC kľúče nasledujúcimi príkazmi:

```
openssl ecparam -genkey -name prime256v1 -noout -out ec_private.pem
openssl ec -in ec_private.pem -pubout -out ec_public.pem
```

Kód 1 Príkazy na vytvorenie EC kľúčov pomocou OpenSSL príkazového riadku

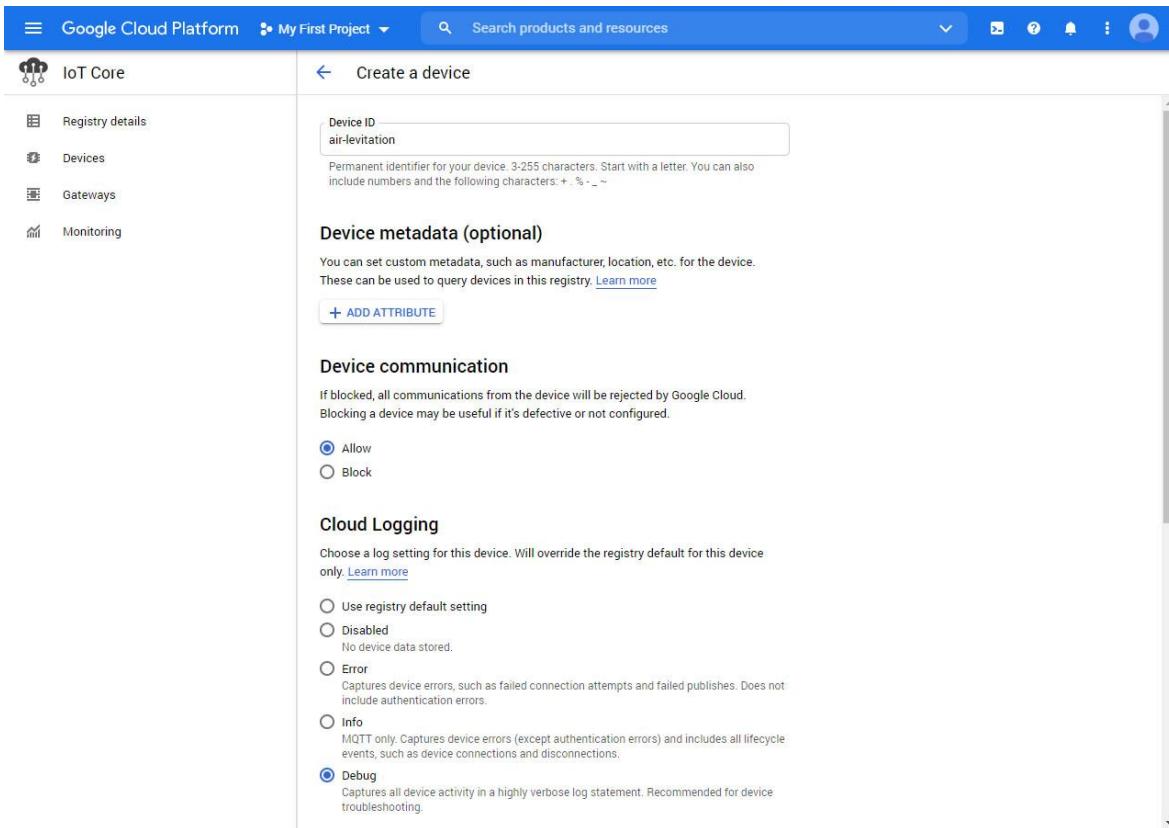
Pomocou druhého príkazu sa nám vytvorí verejný kľúč, ktorý bude potrebné nahrať do GCP. Na to aby sme ho mohli do GCP vložiť, musíme prejsť v rámci GCP na BigData – IoT Core a tam si dať vytvoriť register. Register nazveme „air-levitation-registry“ a ako región

si zvolíme „europe-west1“. Cloud Pub/Sub Topics si zvolíme náš vytvorený „air-levitation-pub“ a zvolíme si ho aj pre Device state topic. Ako protokol si zvolíme MQTT a ach chceme zobrazovať logy z komunikácie, nastavíme si Cloud logging na debug.

The screenshot shows the 'Create a registry' page in the Google Cloud Platform. At the top, there's a navigation bar with 'Google Cloud Platform', 'My First Project', a search bar, and various icons. Below the header, the 'IoT Core' section is visible with a 'Create a registry' button. The main area is titled 'Registry properties'. It contains fields for 'Registry ID' (set to 'air-levitation-registry') and 'Region' (set to 'europe-west1'). Below these, a note says 'Determines where data is stored for devices in this registry. Choice is permanent.' Under 'Cloud Pub/Sub topics', there's a dropdown menu set to 'projects/opportune-box-285621/topics/air-levitation-pub'. A note below it says 'Device telemetry events will be published to this topic by default.' There's also a '+ ADDITIONAL TOPIC' button. The 'Device state topic (optional)' section follows, with a dropdown menu set to 'projects/opportune-box-285621/topics/air-levitation-pub'. A note says 'Device state data will be published to your selected topic on a best-effort basis, as well as to the default MQTT state topic (if your devices use MQTT protocol.)'. At the bottom, there's a 'Protocols' section which is currently collapsed.

Obr. 44 Vytvorenie IoT Core registra v GCP

Ked' sme vytvorili register, potrebujeme vytvoriť ešte inštanciu pre samotné zariadenie a to tak, že dáme „create device“. Ako device ID si nastavíme „air-levitation“, komunikáciu zariadenia povolíme, opäť logovanie stavov si zvolíme podľa potreby, no podstatná je autentifikácia, kde si zvolíme možnosť Upload a vyberieme verejný EC kľúč, ktorý sme si vygenerovali.



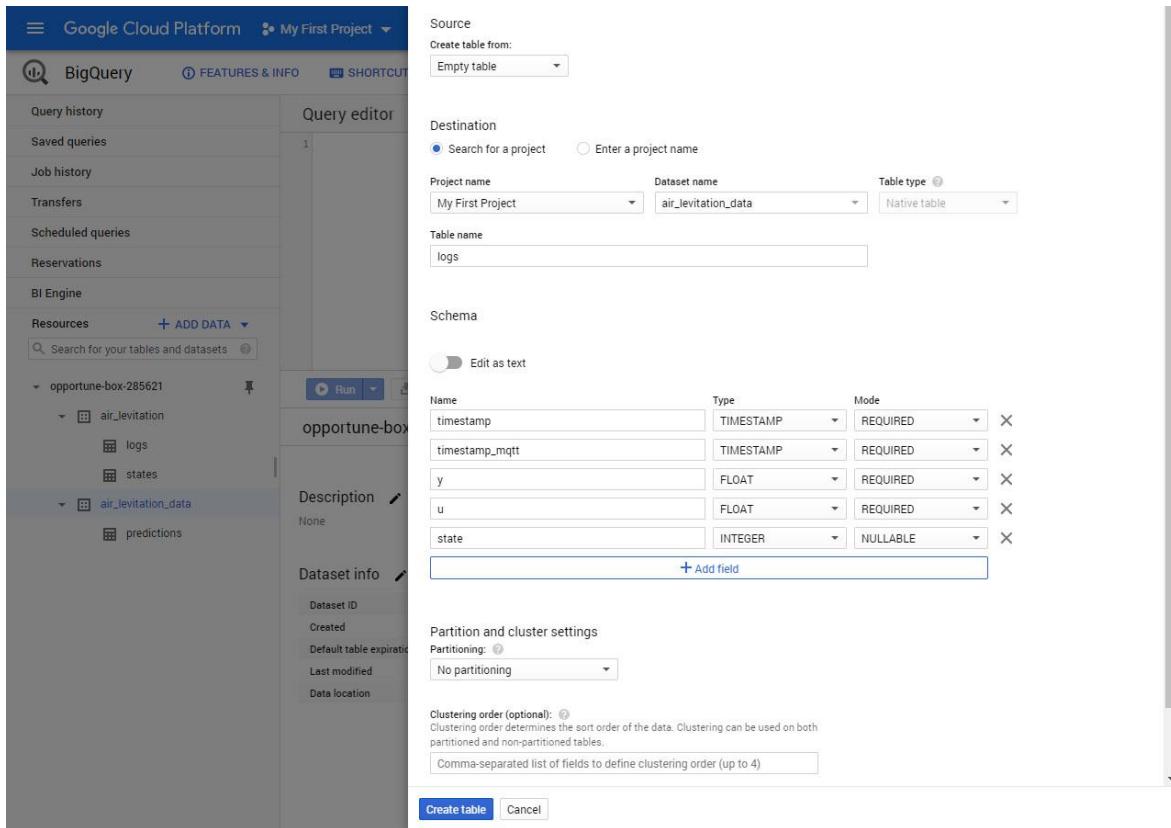
Obr. 45 Registrovanie zariadenia v GCP

Pomocou prvého príkazu sa nám vytvorí privátny kľúč, ktorý bude potrebné otvoriť a jeho obsah prepísat do zdrojových kódov príkladu z knižnice v súbore *ciotc_config.h*. V tomto súbore je potrebné zmeniť všetky parametre, ktoré nastavujú pripojenie na WiFi, informácie z Google IoT Core a znenie samotného privátneho kľúča.

6.3.3 GCP BigQuery

Samotnú komunikáciu zariadenia a clodu máme nastavenú. Ako ďalšie budeme potrebovať databázu našich záznamov. Na ukladanie záznamov nám bude slúži BigQuery. BigQuery záložku nájdeme pod kategóriou BigData. Tu si vytvoríme novú databázu kliknutím na „create dataset“. Ako dataset ID si nastavíme „air_levitation_data“ a lokáciu si zvolíme „europe-west1“. Ostatné položky necháme predvolené.

Po vytvorení nášho datasetu si v ňom vytvoríme novú tabuľku s názvom logs. Táto tabuľka bude obsahovať stĺpce s nastaveniami ako sú na . Ostatné polia ostanú predvolené.



Obr. 46 Vytvorenie tabuľky v databáze BigQuery

Týmto spôsobom si vytvoríme ďalšiu tabuľku. Nasledujúca tabuľka nám bude slúžiť na zachytávanie predikcií jednotlivých segmentov, aby sme mohli naprieč nimi prechádzať a priemerovať ich hodnoty, pričom priemernú hodnotu konkrétneho časového bodu zapíšeme do tabuľky logs. Tabuľku nazveme „predictions“ a budeme v nej uchovávať nasledovné stĺpce:

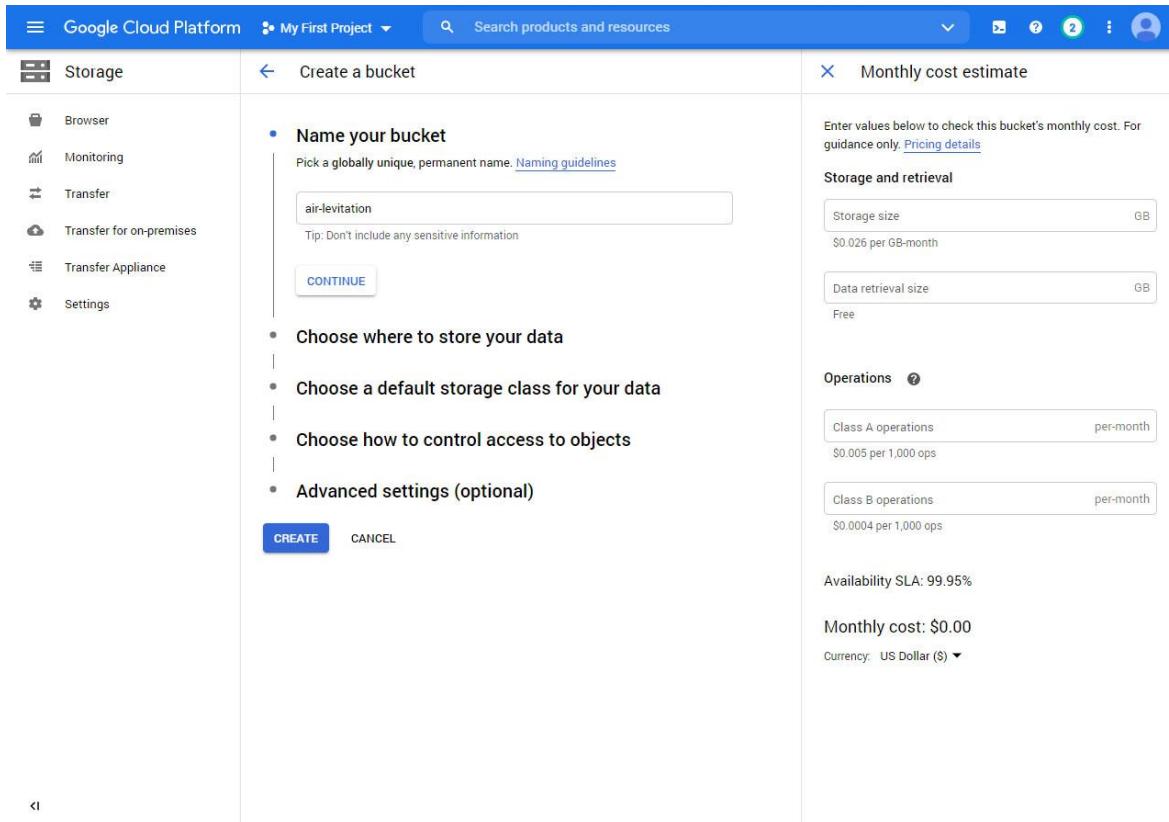
Tabuľka 7 Schéma tabuľky obsahujúca predikcie pre jednotlivé časové body

Názov stĺpca	Typ	Mód
timestamp	TIMESTAMP	REQUIRED
value_	INTEGER	NULLABLE

6.3.4 GCP Storage – Bucket

Na to, aby sme mohli nahrať na GCP svoj natrénovaný model a všetky potrebné pre-rekvizity, si budeme musieť k tomu vytvoriť úložný priestor. Je potrebné si aktivovať Storage, ktorý

sa nachádza pod záložkou Storage. Následne je potrebné vytvoriť bucket, ktorý nazveme „air-levitation“, región zvolíme ako „europe-west1“, trieda úložiska ostane ako štandardná a ostatné vlastnosti necháme predvolené.

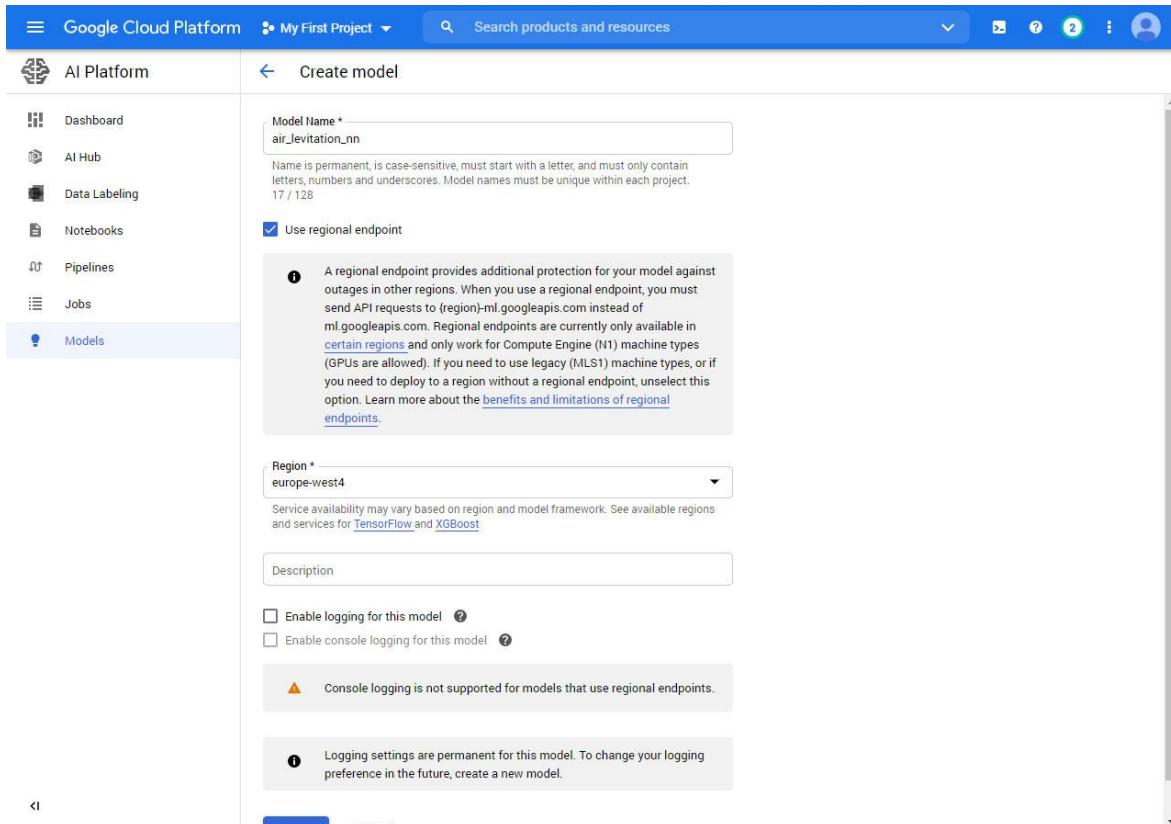


Obr. 47 Vytvorenie bucketu na GCP

Do vytvoreného bucketu následne nakopírujeme obsah prílohy s názvom „air-levitation-bucket“. Podstatnými súbormi sú predovšetkým „nn_predictor-0.1.tar“, v ktorom sa nachádzajú všetky zabalené vlastné súbory potrebné na predikciu a celá zložka „model-dir“, kde sa nachádza samotný natrénovaný model.

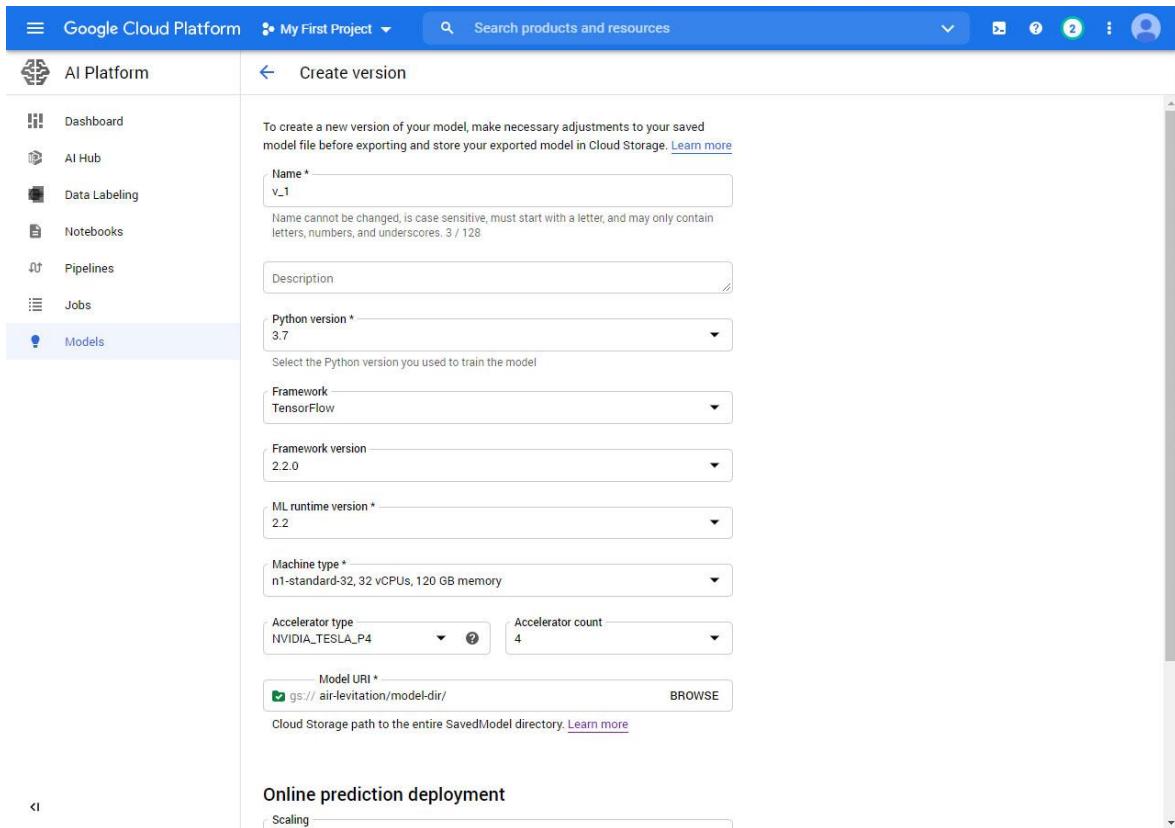
6.3.5 GCP AI Platform

Na to, aby sme vytvorili model v GCP, ktorý nám bude priebežne robiť predikcie vždy, keď bude dopytovaný, je potrebné si tento model najprv v GCP deklarovať. V sekcií Artificial Intelligence, v záložke AI Platform je potrebné si zvoliť položku Models. Následne je potrebné nový model vytvoriť. Názov modelu je „air_levitation_nn“ a región je „europe-west4“.



Obr. 48 Vytvorenie modelu

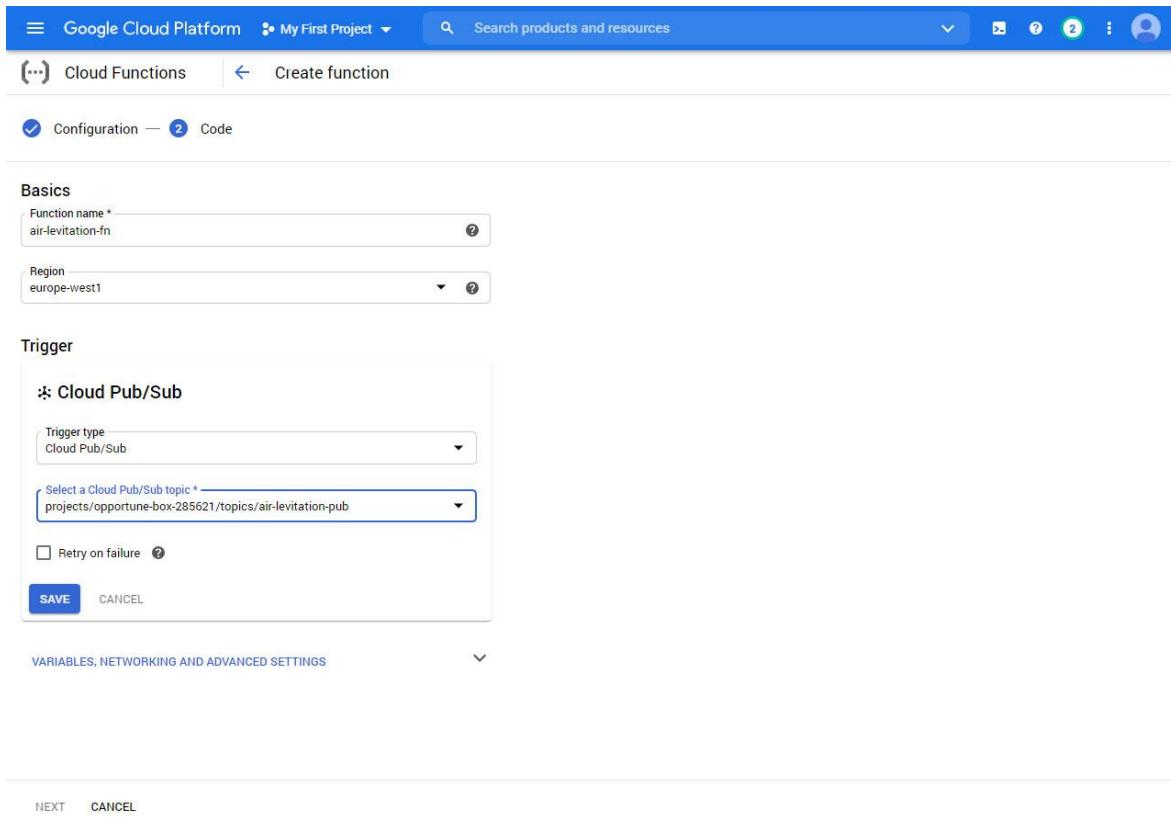
Po vytvorení modelu je potrebné mu definovať verziu. Je vhodné nazývať jednotlivé verzie adekvátnym označením, či už číselným označením alebo podľa majoritnej zmeny. Neodporúčajú sa dlhé názvy. Definujeme verziu pythonu ako 3.7, framework tensorflow 2.2. Nastavenie typu mašiny na predikovanie, počtu CPU, pamäte či typu akcelerátora je skôr otázka preferencií, finančných možností a požadovanej rýchlosťi spracovania požiadavky.



Obr. 49 Vytvorenie verzie modelu

6.3.6 GCP Cloud functions

Po nastavení jednotlivých komponentov, je potrebné ich prepojiť a oživiť komunikáciu. Na to nám slúžia Cloud functions. Je to FaaS – Funkcia ako servis. Nie je potrebné si rezervovať zdroje virtuálneho ani fyzického servera a ani nič s ním spojené. Jediné čo je potrebné a je možné využiť, sú samostatné funkcie, ktoré si môžeme naprogramovať. Pre kontinualitu projektu a vzhľadom na skutočnosť, že neurónové siete sú napísane v jazyku python, tak aj tieto funkcie sú v tomto jazyku spúšťané. Aby sme si ich vytvorili, tak ich najprv dohľadáme pod menu položkou „Compute“. Dáme si vytvoriť novú funkciu, nazveme ju ako „air-levitation-fn“, región nastavíme na „europe-west1“, ako spúšťač bude odoslanie správy na určitú tému a ňou bude „air-levitation-pub“, ktorú sme si vytvorili v časti 6.3.1.



Obr. 50 Vytvorenie udalosťou spúšťanej funkcie v prostredí GCP

Nastavenie vyvolávača funkcie uložíme a pokračujeme v nastaveniach ďalej. V inline editore si navolíme ako jazyk Python 3.7, pričom vstupná funkcia sa volá „`serialize_fn`“. Do tejto časti si nahráme súbory z prílohy „`air-levitation-fn`“. Nachádza sa v nich hlavný kód ktorý sa spúšťa pri prijatí správ zo zariadenia. Dáta sa rozbalia, uložia do databázy, z databázy sa vyberú najnovšie dátá o veľkosti predikovaného okna, tie sa odošlú na predikciu pre umelú inteligenciu. Výsledok umelej inteligencie sa opäť rozbalí a aktualizujú sa tabuľky v databáze. Tieto riadiace hodnoty sú definované v hlavičke súboru s funkciou na spúšťanie udalostí.

7 Testovanie navrhnutého riešenia a zhodnotenie výsledkov

V siedmej kapitole dizertačnej práce sa venujeme trénovaniu a testovaniu navrhнутej architektúry na segmentovanie viac-signálových vstupov, ktoré vstupy, princíp učenia aj predikcie a výstupy sú definované v predošej kapitole. Analyzujeme vplyv nastavenia parametrov architektúry a vyhodnocujeme úspešnosť navrhнутého algoritmu. Na záver analyzujeme časovú náročnosť spracovania dát v jednotlivých krokoch.

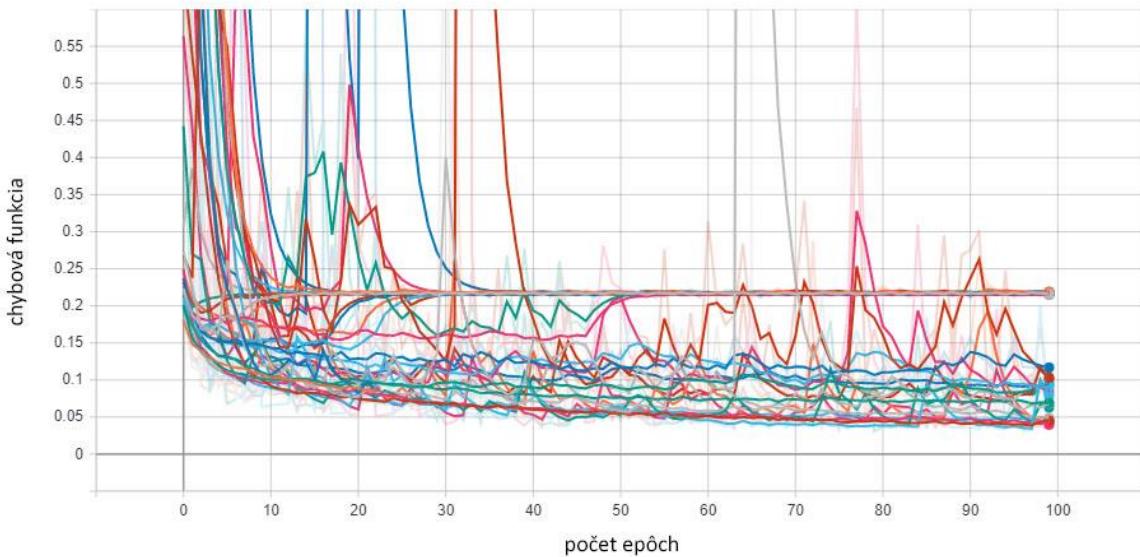
Navrhovanú architektúru neurónovej siete sme natrénovali a otestovali na 30 rôznych nastaveniach architektúry za účelom nájdenia čo najpresnejšie vyhodnocujúceho algoritmu a otestovania aplikatelnosti na definovaný problém. Siet' sme ladili za pomoci inteligentného vyhľadávania hyperparametrov Ax search, založenom na Bayesovom optimalizačnom algoritme. Nastavenia architektúry boli vyberané z definovaných hladín ako sú uvedené v Tabuľka 8. Použilo sa učenie v dávkach, kde bolo 16 vzoriek na dávku, 100 krokov na epochu a 100 epoch na tréning.

Tabuľka 8 Prvé spustenie hľadania hyperparametrov

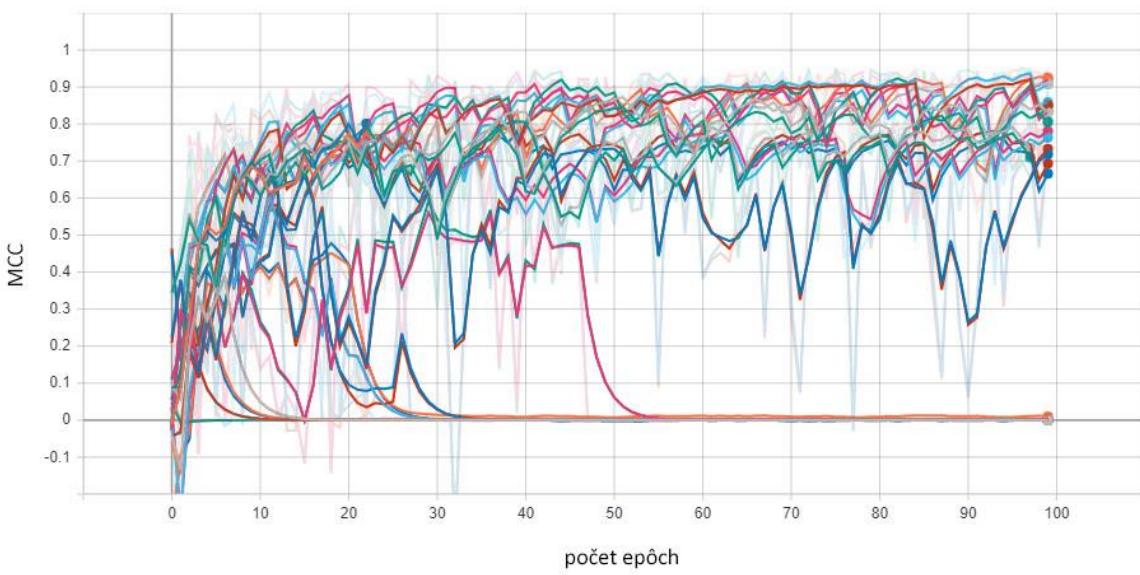
Hyperparameter	Minimálna hodnota	Maximálna hodnota
Velkosť konvolučného filtra	16	32
Počet filtrov v prvej vrstve	32	48
Rýchlosť učenia	0.0001	0.1
Dropout	0.01	0.2
Zmena dropoutu do hĺbky	0.001	0.1
Hĺbka siete	2	4

Počiatok hľadania parametrov prinieslo 10 rôznych nastavení architektúry, podľa ktorých bolo možné spozorovať jej určité správanie. Prvým viditeľným znakom bolo, že vysoká rýchlosť učenia spôsobovala, že siet' po niekoľkých epochách zlepšovania padla vo svojej efektivite na 0 a začala určovať triedu predikcie úplne náhodne, čo opisovala nulová hodnota MCC metriky. Samotná chybová funkcia taktiež stúpla a ustálila sa v okolí hladiny 0.2. Tento fenomén zachytávajú obrázky Obr. 51 Priebeh chybovej funkcie naprieč 100 epochami pre prvých 10 rôznych nastavení a Obr. 52 Priebeh vývoja MCC metriky naprieč 100 epochami pre prvých 10 rôznych nastavení. Ďalším znakom bol takmer nulový rozdiel

v efektivite medzi architektúrou, ktorá mala hĺbku siete 4 a hĺbku siete 3. Rozdiel bol však viditeľný v rýchlosťi učenia. Medzi hĺbkou 2 a 3 už rozdiely badateľné boli a siet' o hĺbke 3 dosahovala lepšie výsledky.



Obr. 51 Priebeh chybovej funkcie naprieč 100 epochami pre prvých 10 rôznych nastavení



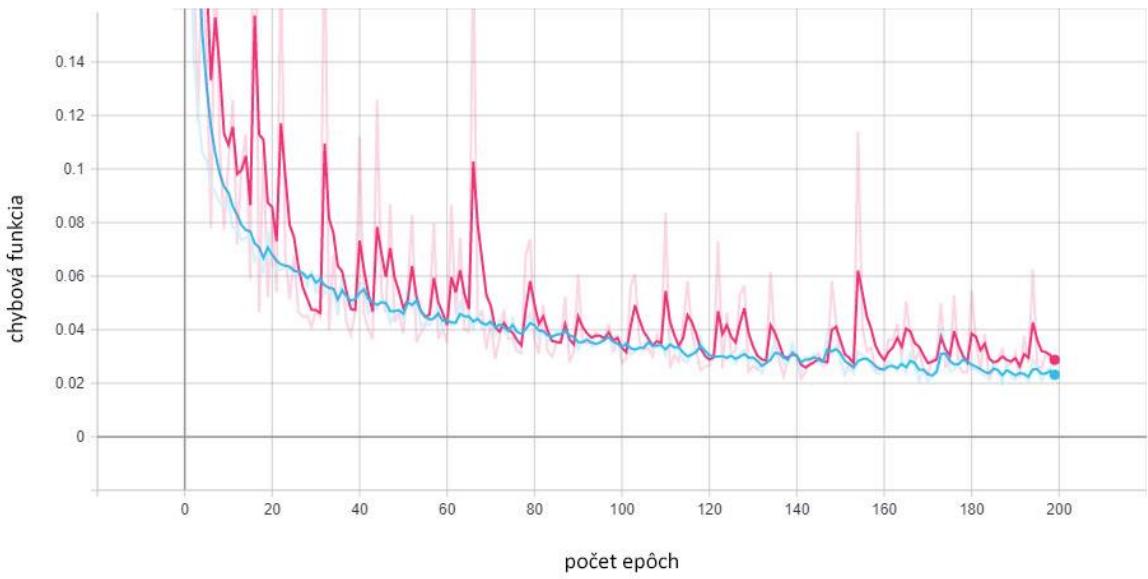
Obr. 52 Priebeh vývoja MCC metriky naprieč 100 epochami pre prvých 10 rôznych nastavení

Vzhľadom na dosahované výsledky bol priestor hyperparametrov upravený tak, ako je zobrazený v Tabuľka 9 Druhé spustenie hľadania hyperparametrov. Hĺbka siete bola fixne definovaná ako hodnota 3.

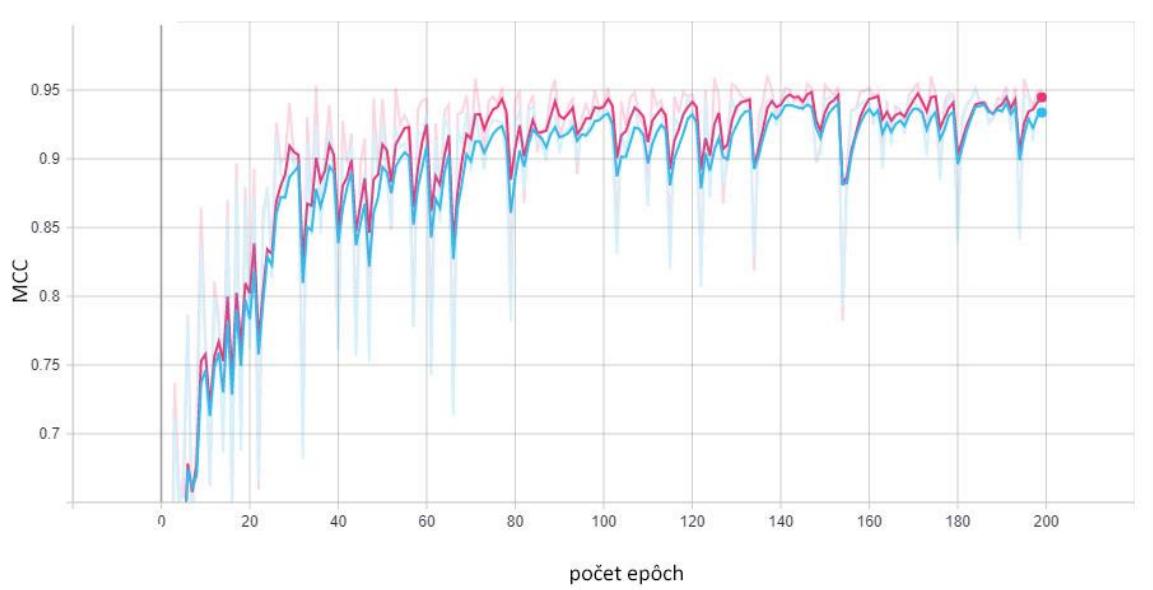
Tabuľka 9 Druhé spustenie hľadania hyperparametrov

<i>Hyperparameter</i>	<i>Minimálna hodnota</i>	<i>Maximálna hodnota</i>
<i>Velkosť konvolučného filtra</i>	16	32
<i>Počet filtrov v prvej vrstve</i>	32	48
<i>Rýchlosť učenia</i>	0.00001	0.01
<i>Dropout</i>	0.01	0.2
<i>Zmena dropoutu do hĺbky</i>	0.0001	0.01

Rádovým znížením hladín rýchlosťi učenia a spomalenia dropoutu do hĺbky sa úplne eliminoval problém modelov s náhodnou predikciou. Modelov v tejto konštelácii bolo vygenerovaných 20. Sledovaná hodnota bola maximálna hodnota validačnej metriky MCC v priebehu trénovala. Na základe tejto hodnoty bolo vybraných 5 najlepších modelov. Všetky dosahovali hodnotu MCC na validačnej vzorke v rozmedzí od 0,96 po 0,964. Po preskúmaní krvky chybovej funkcie bol viditeľný trend stúpania chybovej funkcie na validačnej vzorke v okolí 140 epochy v 3 prípadoch z 5 najlepších modelov. V takomto prípade už dochádza k pretrénovaniu modelu a model samotný už viac nie je schopný sa zlepšovať, preto bola vybraná konfigurácia, ktorá dosahovala v rámci hľadania hyperparametrov nižšie hodnoty, avšak aj po 200 epochách sa ešte stále model zlepšoval.



Obr. 53 Chybová funkcia vybraného modelu po druhom kole hľadaní hyperparametrov



Obr. 54 MCC metrika vybraného modelu po druhom kole hľadaní hyperparametrov

Takto vybraný model bol následne natrénovaný na 500 epochách. Aj keď sa na prvý pohľad zdalo, že by model mohol časom dosahovať lepšie výsledky, po 300 epochách sa začal pretrénovať, do epochy 300 sa pohyboval v podobných hladinách a neskôr jeho chybová funkcia na validačných dátach začala rásť.

Model pre predikciu chýb na zariadení FloatShield pozostáva celkom z 14,737,928 parametrov, z ktorých je 14,734,408 trénovateľných parametrov a 3,520 nie sú trénovateľnými parametrami. Keďže model pre predikciu chýb na zariadení MotoShield

sleduje dvakrát toľko hodnôt čo model pre FloatShield, tak obsahuje aj dvakrát toľko parametrov. Po natrénovaní segmentačných modelov dosahovali najvyššie validačné hodnoty MCC nasledovne:

Tabuľka 10 Výsledky MCC evaluačných metrík získaných modelov na testovacích dátach

<i>xAxis U-Net</i>	<i>Trénovacie MCC</i>	<i>Validačné MCC</i>
<i>FloatShield</i>	0.9467	0.9659
<i>MotoShield</i>	0.9808	0.9676

Vybraný a natrénovaný model pre FloatShield a MotoShield boli otestované na testovacích dátach. Pre bližšiu špecifikáciu výsledkov sa sledovalo klasifikačné skóre pozostávajúce z vlastností akými sú presnosť, sensitivita, F1 skóre, správnosť, makro priemer a vážený priemer. Presnosť je definovaná ako percentuálne vyjadrenie správne určených predikcií ako takých, to znamená, že správne model predikoval pozitívny aj negatívny výskyt danej triedy. Sensitivita je percentuálny podiel iba správne určených pozitívnych predikcií. F1 skóre sleduje vážený harmonický priemer presnosti a sensitivity, čiže sa zameriava predovšetkým na určenie správnych predikcií.

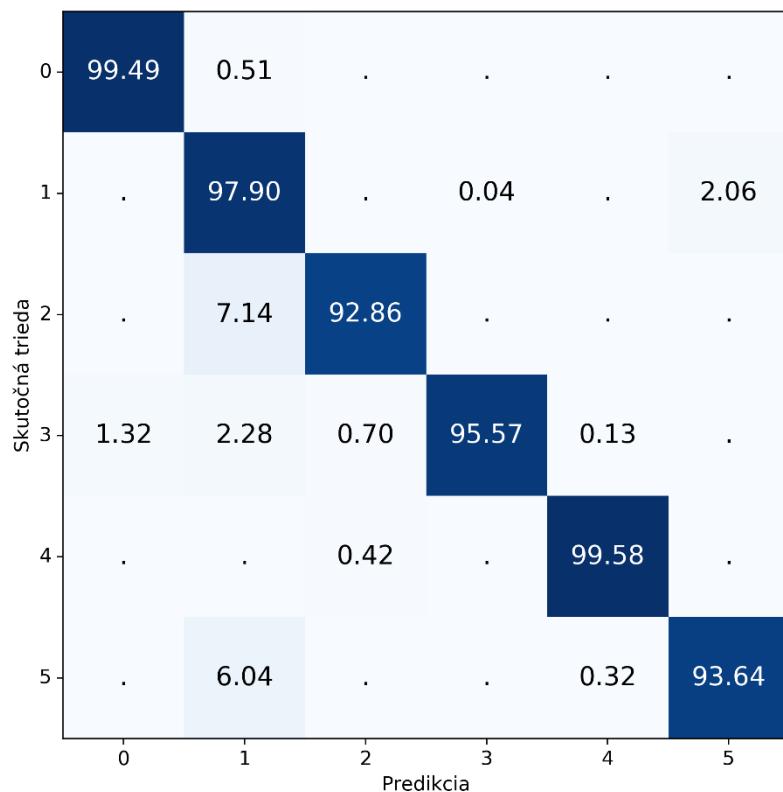
Tabuľka 11 Klasifikačné skóre diagnostického algoritmu pre FloatShield

	<i>Presnosť</i>	<i>Sensitivita</i>	<i>F1 skóre</i>	<i>Počet vzoriek</i>
0	0.99	0.99	0.99	29696
1	0.98	0.98	0.98	49664
2	0.96	0.93	0.94	3584
3	1.00	0.96	0.98	19456
4	0.99	1.00	0.99	3584
5	0.80	0.94	0.86	4352
<i>Správnosť</i>			0.98	110336
<i>Makro priemer</i>	0.95	0.97	0.96	110336
<i>Vážený priemer</i>	0.98	0.98	0.98	110336

Na základe Tabuľka 11 Klasifikačné skóre diagnostického algoritmu pre FloatShield je možné badať, že algoritmus diagnostiky zariadenia pre FloatShield má najväčší problém so stavom číslo 5 a ako druhý najhoršie určený stav je stav číslo 2. FloatShield je definovaný 2 premennými a jeho stav je zaraďovaný do jednej zo šiestich kategórií. Stav číslo 5 určuje

pripad, kedy sa loptička uvoľnila zo zaseknutého stavu, avšak otáčky ventilátora sú neprimerane vysoké a postupne sa znižujú. V takomto prípade je loptička vyfúknutá na vrch tuby a nedosahuje želanú výšku. Jedná sa o extrém. V prípade chyby označenej číslom 2 sa jedná o opačný extrém, čo sa týka výšky loptičky. V tomto prípade je loptička tlačená smerom nadol tăžkým telesom a ventilátor postupne navyšuje otáčky, aby loptičku dostal do požadovanej výšky.

Matica zámeny nám zobrazuje podiel správne predikovaných tried stavov po diagonále a odchýlky v ostatných miestach matice. Podiel je uvádzaný v percentoch. Y-ová os nám označuje skutočné triedy stavov, zatiaľ čo x-ová os označuje predikované triedy stavov. Z matice zámeny je možné vyčítať, v ktorých triedach stavov sa najviac algoritmus mylil a za aké iné triedy svoje predikcie nahradzal. V prípade predikcie stavu 2 aj stavu 5 si ich algoritmus najviac mylil so stavom 1, ktorý označuje chybu, kedy do tuby padlo ľahké teleso, ktoré častokrát blokovalo výškový senzor a ventilátor dosahoval nižšie otáčky, než mal. Samotný charakter chýb je značne odlišný a preto je obťažné určiť príčinu tejto zlej predikcie.



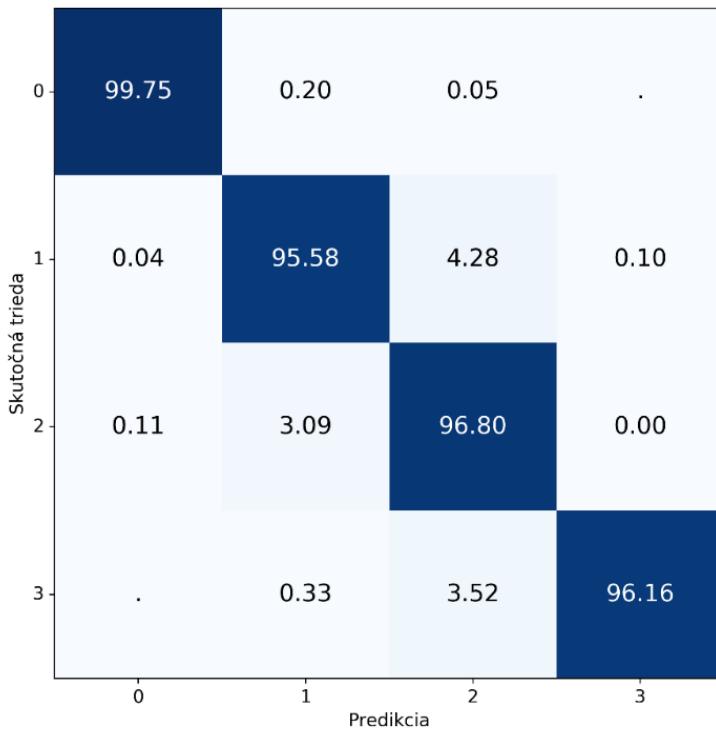
Obr. 55 Matica zámeny pre FloatShield

Vybraný a natrénovaný model pre MotoShield bol otestovaný a vyhodnotený na testovacích dátach. MotoShield má 4 sledované hodnoty a dosahuje 4 rôzne stavy. Tabuľka 12 vyobrazuje klasifikačné skóre diagnostického algoritmu pre MotoShield. Navrhnutá architektúre v prípade MotoShieldu dosahuje rovnaké až lepšie celkové výsledky v každom smere. Dôvodom môže byť väčší objem nameraných dát, väčší počet sledovaných hodnôt ako aj menší počet sledovaných stavov.

Tabuľka 12 Klasifikačné skóre diagnostického algoritmu pre MotoShield

	<i>presnosť'</i>	<i>sensitivita</i>	<i>F1 skóre</i>	<i>Počet vzoriek</i>
<i>0</i>	1.00	1.00	1.00	150528
<i>1</i>	0.96	0.96	0.96	75520
<i>2</i>	0.96	0.97	0.96	75520
<i>3</i>	0.97	0.96	0.97	3072
<i>správnosť'</i>			0.98	304640
<i>Makro priemer</i>	0.97	0.97	0.97	304640
<i>Vážený priemer</i>	0.98	0.98	0.98	304640

Najčastejšia zámena kategórií stavov vyplývajúca z matice zámeny z Obr. 56 Matica zámeny pre MotoShield je pre stav číslo 1. Stav číslo 1 opisuje prípad, kedy cudzie teleso zasahovalo do ozubených koliesok motorčeka. Najčastejšie je tento stav zamieňaný so stavom číslo 2, ktorý predstavuje prípad pôsobenia magnetu rôznej intenzity na rôznych miestach.



Obr. 56 Matica zámeny pre MotoShield

Kroky toku dát sú definované v bodoch s ich priemerným trvaním v sekundách:

- Spustenie funkcie čítania prichádzajúcich dát od poslednej časovej pečiatky (0,003s).
- Prečítanie prichádzajúcich dát (0,288s).
- Vloženie dát do databázy (0,068s).
- Načítanie kľzavého okna za databázy (1,233s).
- Normalizácia dát a ich príprava (0,012s).
- Predikcia stavu v oblasti segmentačného okna (0,505s).
- Transformácia predikcie na tabuľkové dáta (0,012s).
- Uloženie predikcie do databázy (0,176s).

Spracovávanie rýchlosť jednotlivých krokov z celkového hľadiska trvá približne 2,3 sekundy. Tento čas by bolo možné skrátiť v bode predikcie výberom parametrov výkonnejších mašín a v bode načítania kľzavého okna priebežným premazávaním databázy s dočasnými dátami a ukladaním dát zmien do samostatnej databázy.

8 Diskusia

Novodobé informačno-komunikačné technológie v oblasti priemyslu udávajú smer, ktorým sa realizácia výrobných procesov ubera. Postupnou digitalizáciou jednotlivých častí tovární získavame nesmierny objem údajov o jednotlivých procesoch ale predovšetkým o stavoch zariadení. Práve mechatronické zariadenia sú jednou z hlavných základných zložiek výroby a od nich sa odvíjajú mnohé ďalšie postupy. Vhodným diagnostikovaním mechatronického systému je možné optimalizovať výrobné procesy tak, aby dochádzalo k výpadkom vo výrobe čo najmenej a bolo možné včas zachytiť vznikajúce chyby na zariadeniach a tak adekvátnie nastaviť plánovanú údržbu.

V súčasnosti existuje viacero projektov a štúdií, ktoré sa zameriavajú na inteligentnú diagnostiku, avšak najčastejšie sa jednotlivé vedecké výskumy zameriavajú na konkrétnu zariadenia, či sledovanie stavu zariadenia na základe jednej meranej veličiny, čo v mnohých prípadoch nemusí efektívne odhaliť vzniknutú chybu. Spôsob vyhodnocovania stavu zariadenia inteligentnými metódami sa zväčša vykonáva na základe predikcie stavu sledovaného časového úseku ako celku, bez možnosti náhľadu do jednotlivých bodov v čase, pričom jeden časový úsek je vyhodnotený iba raz, čo môže mať za následok zmazávanie a generalizovanie vlastností zariadenia a tým odstránenie skorej detekcie chyby na zariadení.

Z týchto dôvodov má dizertačná práca ako jeden z hlavných dôvodov navrhnúť a implementovať algoritmy intelligentnej diagnostiky mechatronických systémov pre včasné detegovanie chyby na zariadení. Ďalším cieľom dizertačnej práce je návrh a implementácia prepojenia intelligentnej diagnostiky s mechatronickým systémom v rámci konceptu Industry 4.0.

Autorkou navrhnuté algoritmy intelligentnej diagnostiky mechatronických zariadení prezentované v dizertačnej práci sú zovšeobecniteľné a je ich možné aplikovať na mechatronické systémy sledované v čase. Autorkou predstavený segmentačný algoritmus je možné využiť nie len v priemysle, ale v akomkoľvek odvetví, v ktorom je potrebná segmentácia dvojrozmerných vstupných dát po x-ovej alebo y-ovej osi. Takto navrhnutý intelligentný algoritmus postavený na princípoch hlbokého učenia je v rámci konceptu Industry 4.0 implementovaný a overený na modelových zariadeniach. Tieto intelligentné metódy diagnostiky mechatronických systémov sú prepojené s reálnym zariadením

a decentralizovanou výpočtovou jednotkou umiestnenou v cloude na základe autorkou navrhnutom komunikačnom koncepte.

V rámci narastajúcich požiadaviek na optimalizáciu výroby je záujem o inteligentné metódy diagnostiky zariadení a systémov čoraz väčší a ich včasné a úspešné vyhodnotenie ovplyvňuje celý chod výroby. Včasným detegovaním anomalií správania zariadenia je možné predísť neplánovaným odstávkam, je možné zníženie dopadu vzniknutej chyby na zariadenie a tak predísť kritickým situáciám. V kontraste s riešeniami poskytujúcimi identifikáciu a klasifikáciu stavu zariadenia v zhľukoch poskytuje riešenie predstavené v dizertačnej práci možnosti odhalenia chýb, ktoré sa za konvenčných podmienok môžu objaviť až v prípade, kedy táto chyba zasiahla systém dlhodobo, o mnoho skôr a na základe diagnostiky navrhнутej autorkou práce je možné sledovať a pozorovať vzory správania zariadenia, ktoré k dlhodobej chybe smerujú.

Navrhnutý princíp inteligentnej diagnostiky stavu mechatronických zariadení implementovaný do navrhnutého konceptu modernej komunikácie s výpočtovou jednotkou, ktorý je prezentovaný v rámci tejto dizertačnej práce napomáha rýchlejšiemu odhaleniu chyby zariadení a ich vzoroch ich postupného nástupu a výskytu.

Záver

Predložená dizertačná práca sa zaobrá aktuálnou problematikou diagnostiky stavu zariadenia a vhodného zareagovania v kritických prípadoch v kontexte Industry 4.0. Výsledkom práce je modulárne riešenie diagnostikovania stavu zariadenia v online priestore a včasná detekcia chyby na zariadení. Riešenie dizertačnej práce sa nespolieha na dávkové predikovanie stavu zariadenia, ale zameriava sa na jemnejšie spôsoby predikcie. Práca využíva moderný inteligentný prístup ku komunikácii zariadenia s výpočtovou jednotkou a analýzou nelineárnych spojitosťí získaných zo zariadenia v každom bode času. Dizertačná práca je zložená zo siedmych kapitol.

Prínos dizertačnej práce spočíva v návrhu, vývoji a overení metodiky, inteligentných metód a algoritmov diagnostiky technologických zariadení, prostredníctvom najnovších architektúr hlbokých neurónových sietí. Autorka dizertačnej práce pomocou najnovších metód učenia a architektúry hlbokých neurnových sietí navrhla vlastný inteligentný systém diagnostiky, modifikovala tieto metódy a vylepšila pôvodné algoritmy modelovania stavov zariadení a procesov, tak boli využiteľné pre aplikácie v moderných priemyselných procesoch pre úlohy prediktívnej údržby. Oproti konvenčným postupom diagnostiky navrhovaný inteligentný prístup umožňuje na základe intelligentnej detailnej analýzy stavov jednotlivých zariadení vo výrobných a technologických procesoch analyzovať ich stav, predikovať údržbu a tým optimálne dohliadať nad poruchovosťou zariadení. Práca využíva moderný prístup ku komunikácii zariadenia s výpočtovou jednotkou a analýzou nelineárnych spojitosťí vlastností získaných zo zariadenia v každom bode času.

Úvodné tri kapitoly sa zameriavajú na prieskum najnovších technológií v odvetví a ich budúcemu smerovaniu. Prvá kapitola sa zameriava na najnovšie trendy v oblasti informačno-komunikačných technológií v priemysle v znení paradigmy Industry 4.0. Dôraz kladie na koncept a prepojenie jednotlivých vlastností tohto najnovšieho smerovania technológií vo výrobe. Druhá kapitola sa zameriava na umelú inteligenciu a umelé neurónové siete. Vymedzuje základné pojmy a princípy inteligentných technológií v teórii aj v praxi. Zameriava sa na pochopenie komplexu hlbokého učenia, vhodného pre aplikovanie inteligentných technológií v oblasti BigData. Tretia kapitola sa venuje výskumným činnostiam za posledných 5 rokov v oblasti intelligentnej diagnostiky stavu rôznorodých

zariadení. Smeruje svoju pozornosť na spôsoby a aplikácie umelých neurónových sietí na detegovanie chyby zariadenia.

Na tretiu kapitolu nadväzuje kapitola štvrtá, ktorá pojednáva o cieľoch práce, je formulovaný problém, ktorý práca rieši. Je predstavený základný koncept komunikácie zariadenia s výpočtovou jednotkou a inteligentnej diagnostiky zariadenia.

Piata až siedma kapitola tvoria ďažisko dizertačnej práce, v rámci ktorých je uvedený originálny prístup a pracovné postupy, ako aj dosiahnuté výsledky práce. V rámci piatej kapitoly je prezentovaný autorkou vytvorený návrh systému monitorovania a riadenia mechatronických systémov s dôrazom na inteligentnú diagnostiku stavu zariadenia a vhodného zareagovania na prípadné vzniknuté kritické chyby. Návrh pozostáva z definovania požiadaviek na systém a následne navrhnutého konceptu riešenia rozdeleného na tri úrovne: úroveň zariadenia, úroveň výpočtovej jednotky – moderného decentralizovaného prístupu cez clouдовé služby a napokon úroveň prepojenia jednotlivých komponentov v rámci clodu aj mimo neho. Šiesta kapitola sa venuje opisu implementácie riešenia opísaného v dizertačnej práci. Autorkou navrhnutý spôsob komunikácie je implementovaný a predvedený na laboratórnych zariadeniach na výučbu ovládania a fungovania mechatronických systémov vzduchovej levitácie a motorčeka. Tieto zariadenia sú opísané po stránke konštrukcie, ovládania ako aj prepojenia so vzdialenosťou výpočtovou jednotkou pomocou bezdrôtovej komunikácie. Šiesta kapitola ďalej opisuje autorkou navrhnuté algoritmy inteligentného vyhodnocovania stavu zariadenia na základe viacerých nelineárne súvisiacich sledovaných hodnôt v čase. Definuje charakter získaných údajov zo zariadení, autorkou navrhnutej architektúry neurónovej siete tak, aby splňala podmienky jemného vyhodnocovania stavu zariadenia. Tento spôsob vyhodnocovania je definovaný na konci sekcie inteligentnej diagnostiky šiestej kapitoly. Posledná sekcia šiestej kapitoly sa venuje konfigurácii cloudových služieb, databáze údajov, prepojenia jednotlivých komponentov a autorkou navrhnutým algoritmom na obsluhovanie prichádzajúcich údajov zo zariadenia. Siedma kapitola sa venuje overeniu princípov navrhnutých algoritmov v offline aj v online priestore – a teda v produkčnom nasadení. Algoritmy sú testované a ich úspešnosť vyhodnocovaná na príkladoch. Na základe dosiahnutých výsledkov je možné konštatovať, že implementovaním metód intelligentnej diagnostiky je možné včas podchýtiť a reagovať na vznikajúce či vzniknuté problémy spojené s behom zariadenia.

Výsledky dizertačnej práce sú prezentované v siedmej kapitole a predstavujú vedecký a aplikačný prínos k rozvoju študijného odboru kybernetika a mechatronika. Vedecko-aplikačné prínosy dizertačnej práce sú založené na návrhu, realizácii a verifikovaní systému kontinuálnej hĺbkovej inteligentnej diagnostiky zariadenia ako aj samotných algoritmov umelej inteligencie na rozoznávanie jednotlivých vlastností opisujúcich rôzne stavy zariadenia a ich aplikovaní v diagnostickom procese zdravia zariadenia. Prezentované autorkou navrhnuté metódy a postupy definované v dizertačnej práci využívajú princípy konceptu digitalizácie výroby Industry 4.0.

Prínosy dizertačnej práce je možné zhrnúť do nasledovných bodov:

- **Návrh metodiky komunikácie zariadenia s výpočtovou jednotkou v súlade s paradigmou Industry 4.0.**
- **Návrh vlastných inteligentných algoritmov na báze metód hlbokého učenia pre kontinuálnu diagnostiku zariadenia.**
- **Programová realizácia a implementácia navrhnutej metodiky komunikácie zariadenia a intelligentnej jednotky v prostredí cloutu.**
- **Overenie efektivity navrhnutých inteligentných diagnostických algoritmov na báze hlbokého učenia ako aj navrhnutého a implementovaného online systému diagnostiky na dvoch mechatronických laboratórnych fyzikálnych modeloch.**

Dané prínosy dizertačnej práce definované v predošlých bodech predstavujú posledné trendy v oblasti výskumu, vývoja a aplikovania nových postupov a prístupov monitorovania a riadenia mechatronických systémov na báze umelej inteligencie so zameraním na diagnostické postupy do praxe. Výsledky práce sú zovšeobecniteľné a je možné ich ďalej modifikovať pre ďalší výskum a technickú prax aj pre iné odvetvia priemyslu.

Publikačná činnosť autora

ADF Vedecké práce v ostatných domácich časopisoch

- **ADF01** KÉPEŠIOVÁ, Zuzana - KOZÁK, Štefan. An effective face detection algorithm for client-side web-based solutions. In Information Technology Applications = Aplikácie informačných technológií. Vol. 7, No. 2 (2018), s. 43-53. ISSN 1338-6468.
- **ADF02** KÉPEŠIOVÁ, Zuzana - KOZÁK, Štefan. Comparison of regularization and optimization methods for process recognition with use of deep neural network. In Information Technology Applications. Vol. 8, No. 2 (2019), s. 5-13. ISSN 1338-6468.
- **ADF03** KÉPEŠIOVÁ, Zuzana - CIGÁNEK, Ján - KOZÁK, Štefan. Detecting awareness state of a driver during driving. In Information Technology Applications. Vol. 8, No. 2 (2019), s. 45-56. ISSN 1338-6468.

AFC Publikované príspevky na zahraničných vedeckých konferenciách

- **AFC01** KÉPEŠIOVÁ, Zuzana - ROSINOVÁ, Danica - KOZÁK, Štefan. Comparison of optimization techniques for process recognition using deep neural network. In ACCS/PEIT 2019 : 6th International conference on advanced control circuits and systems; 5th International conference on new paradigms in electronics & information technology. Hurghada, Egypt. November 17-20, 2019. Danvers : IEEE, 2019, S. 191-196. ISBN 978-1-7281-6354-3. V databáze: IEEE: 9062866 ; DOI: 10.1109/ACCS-PEIT48329.2019.9062866.
- **AFC02** KÉPEŠIOVÁ, Zuzana - CIGÁNEK, Ján - KOZÁK, Štefan. Driver drowsiness detection using convolutional neural networks. In 2020 Cybernetics & Informatics (K&I) [elektronický zdroj] : 30th International Conference. Velké Karlovice, Czech Republic. January 29-February 1, 2020. 1. ed. Danvers : IEEE, 2020, [6] s. ISBN 978-1-7281-4381-1. V databáze: IEEE: 9039851 ; SCOPUS: 2-s2.0-85083114701 ; DOI: 10.1109/KI48306.2020.9039851.
- **AFC03** CIGÁNEK, Ján - KÉPEŠIOVÁ, Zuzana. Processing and visualization of medical images using machine learning and virtual reality. In 2020 Cybernetics & Informatics (K&I) [elektronický zdroj] : 30th International Conference. Velké Karlovice, Czech Republic. January 29-February 1, 2020. 1. ed. Danvers : IEEE,

2020, [6] s. ISBN 978-1-7281-4381-1. V databáze: IEEE: 9039896 ; SCOPUS: 2-s2.0-85083113151 ; DOI: 10.1109/KI48306.2020.9039896.

AFD Publikované príspevky na domácich vedeckých konferenciách

- **AFD01** KÉPEŠIOVÁ, Zuzana - KOZÁK, Štefan. An effective face detection algorithm. In 2018 Cybernetics & Informatics (K&I) [elektronický zdroj] : 29th International Conference. Lazy pod Makytou, Slovakia. January 31-February 3, 2018. 1. vyd. Bratislava : Slovak Chemical Library, 2018, USB, [6] s. ISBN 978-1-5386-4420-1. V databáze: IEEE: 8337566 ; WOS: 000454633500037 ; SCOPUS: 2-s2.0-85050939647.
- **AFD02** KÉPEŠIOVÁ, Zuzana - ROSINOVÁ, Danica. Application of face detection algorithm on client's side of web solutions by using JavaScript library Tracking.js. In ELITECH'18 [elektronický zdroj] : 20th Conference of doctoral students. Bratislava, Slovakia. May 23, 2018. 1. ed. Bratislava : Vydavateľstvo Spektrum STU, 2018, CD-ROM, [6] p. ISBN 978-80-227-4794-3.
- **AFD03** KÉPEŠIOVÁ, Zuzana - KOZÁK, Štefan. Smart factory in Industry 4.0. In ELITECH'20 [elektronický zdroj] : 22nd Conference of doctoral students. Bratislava, Slovakia. May 27, 2020. 1. ed. Bratislava : Vydavateľstvo Spektrum STU, 2020, [6] s. ISBN 978-80-227-5001-1.
- **AFD04** KÉPEŠIOVÁ, Zuzana. Theoretical proposal of diagnostic and educational tool using digital twin and augmented reality. In ELITECH'19 [elektronický zdroj] : 21st Conference of doctoral students. Bratislava, Slovakia. May 29, 2019. 1. ed. Bratislava : Vydavateľstvo Spektrum STU, 2019, CD-ROM, [7] s. ISBN 978-80-227-4915-2.

Riešiteľka projektov

- 016STU-4/2020 - Virtuálna a zmiešaná realita vo výučbe pre Industry 4.0
- 030STU-4/2017 - Laboratórium digitálnych tovární s podporou IoT

Ostatné

- Vedenie bakalárskych prác: 10ks

Zoznam použitej literatúry

- [1] E. A. Lee, „Cyber Physical Systems: Design Challenges,“ EECS Department, University of California, Berkeley, 1 2008. [Online]. Available: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-8.html>. [Cit. 3 1 2019].
- [2] V. Suwimon a S. Sucha, „INTERNET OF THINGS: A REVIEW OF APPLICATIONS & TECHNOLOGIES,“ *Suranaree Journal of Science and Technology*, zv. 21, %1. vyd.E-ISSN 2587-0009, pp. 359-374, 2014.
- [3] „Industry 4.0 - Priemysel, revolúcia, IoT,“ SOVA Digital a.s., [Online]. Available: <http://industry4.sk/>. [Cit. 3 1 2019].
- [4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica a M. Zaharia, „A view of cloud computing,“ *Communications of the ACM*, zv. 53, %1. vyd.4, pp. 50-58, 2010.
- [5] R. Y.Zhong, X. Xu, E. Klotz a S. T.Newman, „Intelligent Manufacturing in the Context of Industry 4.0: A Review,“ *Engineering*, zv. 3, %1. vyd.5, pp. 616-630, 2017.
- [6] P. Mell a T. Grance, „The NIST Definition of Cloud,“ National Institute of Standards and Technology, Gaithersburg, 2011.
- [7] F. O'Sullivan, „What Is the Internet of Things?,“ 22 6 2018. [Online]. Available: <https://www.cloudwards.net/what-is-the-internet-of-things/>. [Cit. 5 1 2019].
- [8] R. Khan, S. U. Khan, R. Zaheer a S. Khan, „Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges,“ rev. *10th International Conference on Frontiers of Information Technology*, Islamabad, 2012.
- [9] Anusha.M, S. Babu.E, S. M. Reddy.L, V. Krishna.A a Bhagyasree.B, „PERFORMANCE ANALYSIS OF DATA PROTOCOLS OF INTERNET OF THINGS: A QUALITATIVE REVIEW,“ *International Journal of Pure and Applied Mathematics*, zv. 115, %1. vyd.6, pp. 37-47, 2017.
- [10] „What is CoAP IoT protocol | CoAP Architecture,message header,“ [Online]. Available: <http://www.rfwireless-world.com/IoT/CoAP-protocol.html>. [Cit. 1 19 2019].
- [11] W. Wong, „What's the Difference Between Consumer and Industrial IoT?,“ 21 9 2016. [Online]. Available: <https://www.electronicdesign.com/iot/what-s-difference-between-consumer-and-industrial-iot>. [Cit. 5 1 2019].
- [12] W. Kitzinger, M. Karner, G. Traar, J. Henjes a W. Sihn, „Digital Twin in manufacturing: A categorical literature review and classification,“ rev. *16th IFAC Symposium on Information Control Problems in Manufacturing*, Bergamo, 2018.

- [13] „Digital Twin | GE Digital,“ GE Digital, [Online]. Available: <https://www.ge.com/digital/applications/digital-twin>. [Cit. 6 1 2019].
- [14] A. Konar, Artificial Intelligence and Soft Computing, Behavioral and Cognitive Modeling of the Human Brain, CRC Press, 2018.
- [15] Y. Pan, „Heading toward Artificial Intelligence 2.0,“ *Engineering*, zv. 2, pp. 409-413, 2016.
- [16] I. Goodfellow, Y. Bengio a A. Courville, Deep Learning (Adaptive Computation and Machine Learning series), The MIT Press, 2016.
- [17] W. Sun, J. Liu a Y. Yue, „AI-Enhanced Offloading in Edge Computing: When Machine Learning Meets Industrial IoT,“ *IEEE Network*, pp. 68-74, 2019.
- [18] A. Ng a K. Katanforoosh, „Deep Learning,“ *CS229 Lecture notes*, pp. 1-16, 2018.
- [19] P. Ramachandran, B. Zoph a Q. V. Le, „Searching for Activation Functions,“ *arXiv:1710.05941*, 2017.
- [20] A. Amidi a S. Amidi, „Deep Learning cheatsheet,“ Standford, 15 September 2018. [Online]. Available: <https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-deep-learning>. [Cit. 13 Január 2020].
- [21] T. Ma, A. Avati, K. Katanforoosh a A. Ng, „Deep Learning,“ *CS229 Lecture notes*, pp. 1-21, 2020.
- [22] A. Ng, Machine Learning Yearning-Draf, deeplearning.ai, 2018.
- [23] F. Van Veen, „The Neural Network ZOO,“ The Asimov Institute, 14 September 2016. [Online]. Available: <https://www.asimovinstitute.org/neural-network-zoo/>. [Cit. 17 10 2019].
- [24] A. Amidi a S. Amidi, „Recurrent Neural Networks cheatsheet,“ Standford, 25 November 2018. [Online]. Available: <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>. [Cit. 6 Marec 2020].
- [25] S. Venkatesan, K. Manickavasagam, N. Tengenkai a N. Vijayalakshmi, „Health monitoring and prognosis of electric vehicle motor using intelligent-digital twin,“ *IET Electric Power Applications*, zv. 13, %1. vyd.9, p. 1328 – 1335, 2019.
- [26] Y. Xu, Y. Sun, X. Liu a Y. Zheng, „A Digital-Twin-Assisted Fault Diagnosis Using Deep Transfer Learning,“ *IEEE Access*, zv. 7, pp. 19990 - 19999, 2019.
- [27] J. Wang, L. Ye, R. X. Gao, C. Li a L. Zhang, „Digital Twin for rotating machinery fault diagnosis in smart manufacturing,“ *International Journal of Production Research*, zv. 57, %1. vyd.12, p. 3920–3934, 2018.
- [28] X. Guo, L. Chen a C. Shen, „Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis,“ *Measurement*, zv. 93, pp. 490-502, 2016.
- [29] W. Zhang, C. Li, G. Peng, Y. Chen a Z. Zhang, „A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy

- environment and different working load,“ *Mechanical Systems and Signal Processing*, zv. 100, pp. 439-453, 2018.
- [30] O. Janssens, V. Slavkovikj, B. Vervisch, K. Stockman, M. Loccufier, S. Verstockt, R. Van de Walle a S. Van Hoecke, „Convolutional Neural Network Based Fault Detection for Rotating Machinery,“ *Journal of Sound and Vibration*, zv. 377, pp. 331-345, 2016.
- [31] L. Wen, X. Li, L. Gao a Y. Zhang, „A New Convolutional Neural Network-Based Data-Driven Fault Diagnosis Method,“ *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, zv. 65, % 1. vyd.7, pp. 5990-5998, 2018.
- [32] H. Liu, J. Zhou, Y. Zheng, W. Jiang a Y. Zhang, „Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders,“ *ISA Transactions*, zv. 77, pp. 167-178, 2018.
- [33] H. Zhao, S. Sun a B. Jin, „Sequential Fault Diagnosis Based on LSTM Neural Network,“ *IEEE Access*, zv. 6, pp. 12929-12939, 2018.
- [34] F. Jia, Y. Lei, L. Guo, J. Lin a S. Xing, „A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines,“ *Neurocomputing*, zv. 272, pp. 619-628, 2018.
- [35] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang a X. Chen, „A sparse auto-encoder-based deep neural network approach for induction motor faults classification,“ *Measurement*, zv. 89, pp. 171-178, 2016.
- [36] S. Haidong, J. Hongkai, Z. Huiwei a W. Fuan, „A novel deep autoencoder feature learning method for rotating machinery fault diagnosis,“ *Mechanical Systems and Signal Processing*, zv. 95, pp. 187-204, 2017.
- [37] G. Takács, „AutomationShield,“ [Online]. Available: <https://github.com/gergelytakacs/AutomationShield/>. [Cit. 12 11 2019].
- [38] T. Gergely, C. Peter, M. Gulán, E. Mikuláš, J. Kulhánek, G. Penzinger, M. Vdoleček, M. Podbielančík, M. Lučan, P. Šálka a D. Šroba, „FloatShield: An Open Source Air Levitation Device for Control Engineering Education,“ rev. *21st IFAC World Congress*, Berlin, 2020.
- [39] O. Ronneberger, P. Fischer a T. Brox, „U-Net: Convolutional Networks for Biomedical Image Segmentation,“ rev. *International Conference on Medical image computing and computer-assisted intervention*, 2015.
- [40] M. Perslev, M. H. Jensen, S. Darkner, P. J. Jenum a C. Igel, „U-Time: A Fully Convolutional Network for Time Series Segmentation Applied to Sleep Staging,“ rev. *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, 2019.
- [41] T.-Y. Lin, . P. Goyal, . R. Girshick, . K. He a P. Dollár, „Focal Loss for Dense Object Detection,“ *IEEE Transactions on Pattern Analysis and Machine Intelligence*, zv. 42, % 1. vyd.2, pp. 318 - 327, 2018.

- [42] G. Jurman, S. Riccadonna a C. Furlanello, „A Comparison of MCC and CEN Error Measures in Multi-Class Prediction,“ *PLoS ONE*, zv. 7, %1. vyd.8, 2012.
- [43] R. Delgado a X.-A. Tibau, „Why Cohen’s Kappa should be avoided as performance measure in classification,“ *PLoS ONE*, zv. 14, %1. vyd.9, 2019.

Prílohy

A. Štruktúra elektronického nosiča II

A. Štruktúra elektronického nosiča

/AI/

/Cloud Functions/

/Device/

práca.pdf